

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Separação Cega de Misturas com Não-Linearidade Posterior Utilizando Estruturas Monotônicas e Algoritmos Bio-Inspirados de Otimização

Autor

Filipe de Oliveira Pereira

Orientador

Prof. Dr. Romis Ribeiro de Faissol Attux

Co-Orientador

Dr. Leonardo Tomazeli Duarte

Banca Examinadora:

Prof. Dr. Romis Ribeiro de Faissol Attux (FEEC/UNICAMP)

Prof. Dr. Magno Teófilo Madeira da Silva (Escola Politécnica/USP)

Prof. Dr. Fernando José Von Zuben (FEEC/UNICAMP)

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Campinas, 19 de agosto de 2010.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

P414s Pereira, Filipe de Oliveira
 Separação cega de misturas com não-linearidade
 posterior utilizando estruturas monotônicas e algoritmos
 bio-inspirados de otimização / Filipe de Oliveira Pereira.
 – Campinas, SP: [s.n.], 2010.

 Orientadores: Romis Ribeiro de Faissol Attux,
 Leonardo Tomazeli Duarte.

 Dissertação de Mestrado - Universidade Estadual de
 Campinas, Faculdade de Engenharia Elétrica e de
 Computação.

 1. Processamento de sinais. 2. Entropia (Teoria da
 informação). 3. Teoria da informação. 4. Sistemas não-
 lineares. 5. Algoritmos genéticos. I. Attux, Romis
 Ribeiro de Faissol. II. Duarte, Leonardo Tomazeli. III.
 Universidade Estadual de Campinas. Faculdade de
 Engenharia Elétrica e de Computação. IV. Título.

Título em Inglês: Blind separation of post-nonlinear mixtures using monotonic
structures and bio – inspired optimization algorithms

Palavras-chave em Inglês: Signal processing, Entropy (Information theory),
Information theory, Nonlinear systems, Genetic algorithms

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca Examinadora: Magno Teófilo Madeira da Silva, Fernando José Von Zuben

Data da defesa: 19/08/2010

Programa de Pós Graduação: Engenharia Elétrica

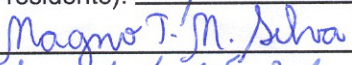
COMISSÃO JULGADORA - TESE DE MESTRADO

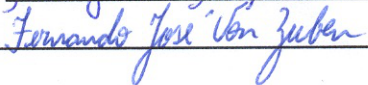
Candidato: Filipe de Oliveira Pereira

Data da Defesa: 19 de agosto de 2010

Título da Tese: "Separação Cega de Misturas com Não-Linearidade Posterior Utilizando Estruturas Monotônicas e Algoritmos Bio-Inspirados de Otimização"

Prof. Dr. Romis Ribeiro de Faissol Attux (Presidente): 

Prof. Dr. Magno Teófilo Madeira da Silva: 

Prof. Dr. Fernando José Von Zuben: 

Resumo

O presente trabalho se propõe a desenvolver métodos de Separação Cega de Fontes (BSS) para modelos de mistura com Não-Linearidade Posterior (PNL). Neste caso particular, a despeito da não-linearidade do modelo, ainda é possível recuperar as fontes através de técnicas de Análise de Componentes Independentes (ICA). No entanto, há duas dificuldades maiores no emprego da ICA em modelos PNL. A primeira delas diz respeito a uma restrição sobre as funções não-lineares presentes no modelo PNL: elas devem ser monotônicas por construção. O segundo problema se encontra no ajuste do sistema separador com base em funções custo associadas à ICA: pode haver mínimos locais sub-ótimos. De modo a contornar o primeiro problema, investigamos a adequabilidade de três tipos distintos de estruturas não-lineares monotônicas. Para lidar com a presença de mínimos sub-ótimos no ajuste do sistema separador, empregamos algoritmos bio-inspirados com significativa capacidade de busca global. Finalmente, buscamos, através de experimentos em diversos cenários representativos, identificar dentre as estratégias estudadas qual a melhor configuração, tanto em termos de qualidade da estimação das fontes quanto em termos de complexidade.

Palavras-chave: separação cega de fontes, misturas com não-linearidade posterior, estruturas monotônicas, computação natural, análise de componentes independentes.

Abstract

This work aims at the development of Blind Source Separation (BSS) methods for Post-NonLinear (PNL) mixing models. In this particular case, despite the presence of nonlinear elements in the mixing model, it is still possible to recover the sources through Independent Component Analysis (ICA) methods. However, there are two major problems in the application of ICA techniques to PNL models. The first one concerns a restriction on the nonlinear functions present in the PNL model: they must be monotonic functions by construction. The second one is related to the adjustment of the PNL separating system via ICA-based cost functions: there may be sub-optimal local minima. To cope with the first problem, we investigate three types of monotonic nonlinear structures. Moreover, to circumvent the problem related to the presence of sub-optimal minima, we consider bio-inspired algorithms that have a significant global search potential. Finally, we perform a set of experiments in representative scenarios in order to identify, among the considered strategies, the best ones in terms of quality of the retrieved sources and overall complexity.

Keywords: blind source separation, post-nonlinear mixtures, monotonic structures, natural computing, independent component analysis.

Agradecimentos

Primeiramente a Deus, por sempre me iluminar e me proteger, por me guiar nos momentos difíceis e me mostrar o caminho da verdade.

Aos meus pais, José Nivaldo e Lázara, por estarem sempre presentes em minha vida, por fazerem dos meus sonhos uma realidade e nunca deixarem de me mostrar a real importância do estudo e da sabedoria, sendo para mim exemplos de força e dignidade, e, também à minha irmã Raquel, pelo apoio.

À minha noiva, Tatiane, por todo o seu amor, sua amizade e por estar sempre ao meu lado, me apoiando e incentivando a sempre buscar coisas novas.

Ao meu orientador, professor Romis Ribeiro de Faissol Attux, por se mostrar além de meu orientador um grande amigo, sempre disposto a me indicar os melhores caminhos para a perfeita elaboração dessa dissertação, me incentivando a buscar o que há de melhor em mim.

Ao meu co-orientador, Leonardo Tomazeli, que me auxiliou em todos os momentos de dificuldades e de dúvidas, mas principalmente pela dedicação e amizade.

Aos amigos do LOsTe: Bixo, Cotonete, Kexada, Marco, Pasté, Rodrigo, Rubão e Tiezzi.

Aos amigos do DSPCom, pelas discussões proveitosas, e também pela amizade e companheirismo em todos os momentos.

Aos membros da banca pelas valiosas sugestões.

À todos que puderam comparecer à defesa.

À toda minha família e aos amigos, pela imensa torcida para que tudo acontecesse da melhor maneira possível.

À CAPES, pelo apoio financeiro.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Abreviaturas	x
Lista de Símbolos	xi
1 Introdução	1
1.1 Organização	2
2 Separação Cega de Fontes	4
2.1 Histórico	4
2.2 Aplicações	5
2.2.1 Separação de Sinais de Áudio - <i>Cocktail-party problem</i>	6
2.2.2 Separação de Imagens	6
2.2.3 Telecomunicações - BSS e Equalização Cega de Canais	7
2.2.4 Arranjo de Sensores Químicos	8
2.3 Definição Matemática do Problema de BSS	8
2.3.1 Sistemas Lineares ou Não-Lineares	9
2.3.2 Sistemas Instantâneos e com Memória	9
2.3.3 Relação entre Número de Fontes e Sensores	9
2.4 Análise de Componentes Independentes	10
2.4.1 Independência Estatística x Descorrelação	11
2.4.2 Separabilidade	12
2.5 Análise de Componentes Independentes - ICA: Critérios e Estratégias	12
2.5.1 Informação Mútua	13
2.5.2 Estimação por Máxima Verossimilhança	14
2.5.3 Infomax	16
2.5.4 Maximização da Não-Gaussianidade - Negentropia, Curtose e FastICA	18
2.5.5 FastICA	19
2.5.6 Análise por Componentes Principais (PCA), Branqueamento e PCA Não-Linear	22
2.6 Separação de Misturas Não-Lineares	25
2.6.1 Misturas Não-Lineares	26

2.6.2	ICA Não-Linear	26
2.7	Mistura com Não-Linearidade Posterior ou <i>Post-Nonlinear</i>	27
2.7.1	Separabilidade do Modelo PNL	28
2.8	O Algoritmo de Taleb e Jutten	29
2.9	Conclusão: Dificuldades em Separação de Misturas PNL	31
3	Otimização da Informação Mútua Utilizando Algoritmos Bio-Inspirados	32
3.1	CrITÉrio de Informação Mútua e Estatísticas de Ordem	33
3.1.1	Estimação de Entropia Utilizando Estatísticas de Ordem	33
3.2	Computação Natural e Otimização: Sistemas Imunológicos Artificiais e Inteligência de Enxame	34
3.2.1	Sistemas Imunológicos Artificiais	35
3.2.2	Princípio da Seleção Clonal	37
3.2.3	Aspectos da Implementação do Algoritmo CLONALG	37
3.3	Otimização Baseada em Enxames de Partículas	39
3.3.1	Aspectos da Implementação do Algoritmo Enxame de Partículas	39
3.4	Conclusões	41
4	Estruturas Monotônicas	42
4.1	Polinômios	42
4.2	Rede Monotônica de Sill	43
4.3	Perceptron de Múltiplas Camadas com Restrição de Monotonicidade	44
4.4	Conclusões	46
5	Resultados	48
5.1	Análise do Emprego de Ferramentas Bio-Inspiradas no Âmbito da Metodologia de Duarte <i>et al.</i> [1]	49
5.2	Análise do Uso de Estruturas Monotônicas no Modelo PNL	53
5.2.1	Resultados – Polinômios	54
5.3	Resultados – Rede Monotônica de Sill	58
5.4	Resultados – Rede Neural MLP Monotônica	64
6	Conclusões e Perspectivas	70

Lista de Figuras

2.1	<i>O cocktail-party problem</i> (retirado de [2])	6
2.2	Imagens Originais	7
2.3	Imagens Misturadas	7
2.4	Esquema de Equalização	7
2.5	Diagrama Esquemático de Separação Cega de Fontes	9
2.6	Estrutura do sistema separador no critério Infomax (retirado de [2])	16
2.7	Estratégias para ICA	21
2.8	Tratamento da BSS considerando estatística de segunda ordem	24
2.9	Modelo PNL	28
3.1	União entre receptores e epítomos (baseado em [3])	36
4.1	Arquitetura da Rede Monotônica de Sill	43
4.2	Arquitetura da uma Rede Neural MLP.	45
4.3	MLP <i>sem</i> Restrição de Monotonicidade X MLP <i>com</i> Restrição de Monotonicidade	46
5.1	Procedimento da Análise de Resultados	49
5.2	Resultados - Primeiro Cenário (CLONALG)	52
5.3	Distribuição dos sinais (a) observados e (b) recuperados.	55
5.4	Composição das funções f_i e g_i para as duas fontes.	56
5.5	Estimativas de cada uma das três fontes.	57
5.6	Distribuição dos sinais (a) observados e (b) recuperados.	59
5.7	Composição das funções f_i e g_i para as duas fontes.	59
5.8	Distribuição dos sinais (a) observados e (b) recuperados.	61
5.9	Composição das funções f_i e g_i para as duas fontes.	62
5.10	Distribuição dos sinais (a) observados e (b) recuperados.	63
5.11	Composição das funções f_i e g_i para as duas fontes.	63
5.12	Distribuição dos sinais (a) observados e (b) recuperados.	65
5.13	Composição das funções f_i e g_i para as duas fontes.	66
5.14	Distribuição dos sinais (a) observados e (b) recuperados.	66
5.15	Composição das funções f_i e g_i para as duas fontes.	67
5.16	Distribuição dos sinais (a) observados e (b) recuperados.	68
5.17	Composição das funções f_i e g_i para as duas fontes.	68

Lista de Tabelas

2.1	Algoritmo FastICA utilizando a Negentropia como critério de otimização	20
2.2	Algoritmo de Taleb e Jutten	30
3.1	Descrição do Algoritmo CLONALG	38
3.2	Descrição do Algoritmo de Enxame de Partículas	40
5.1	Parâmetros do Algoritmo CLONALG	50
5.2	Parâmetros do Algoritmo Enxame de Partícula	50
5.3	EQM - Primeiro Cenário	51
5.4	Parâmetros do Algoritmo CLONALG no Segundo Cenário	52
5.5	Parâmetros do Algoritmo Enxame de Partícula no Segundo Cenário	53
5.6	EQM - Segundo Cenário	53
5.7	Parâmetros do Algoritmo CLONALG	55
5.8	Parâmetros do Algoritmo CLONALG	56
5.9	Parâmetros do Algoritmo CLONALG	58
5.10	Parâmetros do Algoritmo CLONALG	60
5.11	Parâmetros do Algoritmo CLONALG	62
5.12	Parâmetros do Algoritmo CLONALG	64
5.13	Parâmetros do Algoritmo CLONALG	65
5.14	Parâmetros do Algoritmo CLONALG	67

Lista de Abreviaturas

<i>AIS</i>	- <i>Artificial Immune System</i> - Sistema Imunológico Artificial
<i>BSS</i>	- <i>Blind Source Separation</i> - Separação Cega de Fontes
<i>dB</i>	- decibel
<i>EQM</i>	- Erro Quadrático Médio
<i>ICA</i>	- <i>Independent Component Analysis</i> - Análise de Componentes Independentes
<i>Infomax</i>	- <i>Information Maximization</i>
<i>MIMO</i>	- <i>Multiple-Input Multiple-Output</i>
<i>MLP</i>	- <i>Mutli-Layer Perceptron</i> - Perceptron de Múltiplas Camadas
<i>MMSE</i>	- <i>Minimum Mean Square Error</i> - Erro Quadrático Médio Mínimo
<i>MONMLP</i>	- <i>Monotonic Multi-Layer Perceptron</i> - Perceptron de Múltiplas Camadas Monotônico
<i>NPCA</i>	- <i>Nonlinear Principal Component Analysis</i> - Análise de Componentes Principais Não-linear
<i>PCA</i>	- <i>Principal Component Analysis</i> - Análise de Componentes Principais
<i>pH</i>	- Potencial Hidrogeniônico
<i>PNL</i>	- <i>Post-Nonlinear</i>
<i>PS</i>	- <i>Particle Swarm</i> - Enxame de Partículas
<i>PSO</i>	- <i>Particle Swarm Optimization</i> - Otimização poe Enxame de Partículas
<i>SIR</i>	- <i>Signal-to-Interference Ratio</i> - Relação Sinal-Interferência
<i>SISO</i>	- <i>Single-Input Single-Output</i>

Lista de Símbolos

a ou A	- escalar
\mathbf{a}	- vetor (letra minúscula e negrito)
\mathbf{A}	- matriz (letra maiúscula em negrito)
\mathbf{I}	- matriz identidade
$p(x)$	- Função densidade de probabilidade da variável aleatória x
$P(x)$	- Função distribuição de probabilidade da variável aleatória x
$p(x, y)$	- Função densidade de probabilidade conjunta das variáveis aleatórias x e y
$f(\cdot), g(\cdot)$	- função
$E_a[\cdot], E_A[\cdot]$	- operador esperança com relação à variável aleatória a ou \mathbf{A}
$\arg \min_{\mathbf{a}}\{f(\cdot)\}$	- argumento \mathbf{a} que minimiza a função $f(\cdot)$
$\arg \max_{\mathbf{a}}\{f(\cdot)\}$	- argumento \mathbf{a} que maximiza a função $f(\cdot)$
$\min(\cdot)$	- mínimo valor de (\cdot)
$\max(\cdot)$	- máximo valor de (\cdot)
$\ \mathbf{a}\ , \ \mathbf{a}\ _2$	- norma 2 do vetor \mathbf{a}
$\sum_{i=c_1}^{c_2}$	- somatório com índice i variando de c_1 a c_2
\sum_i	- somatório com índice i
\log	- logaritmo
$L(\mathbf{w})$	- logaritmo da função de verossimilhança do vetor \mathbf{w}

Capítulo 1

Introdução

Em muitas situações práticas que envolvem a coleta de informações do ambiente, pode ocorrer que os sinais fornecidos pelos sensores sejam misturas de sinais de interesse (fontes). O processo de recuperação desses sinais a partir de um mínimo de informação a priori sobre o sistema misturador dá origem ao problema de Separação Cega de Fontes (BSS). Tal problema pode ser encontrado em diversas aplicações práticas, como processamento de sinais e imagens [4], telecomunicações [5] e análise iônica através de eletrodos íon-seletivos [6].

Um dos métodos mais difundidos para solução do problema de BSS é a Análise de Componentes Independentes (ICA), que assume como hipótese central que as fontes são estatisticamente independentes e não-gaussianas. Assim sendo, o processo de separação através de ICA é feito justamente de modo a recuperar a independência das fontes, que se perde no processo de mistura [7].

Embora o uso de ICA em BSS tenha se consolidado fundamentalmente no âmbito de modelos lineares, a existência de situações práticas nas quais o processo de mistura possui elementos não-lineares vem motivando extensões capazes de operar num contexto mais abrangente. No caso não-linear geral, no entanto, surge uma dificuldade fundamental para o uso de ICA: recuperar a condição de independência pode não levar a uma correta estimação das fontes. Essa dificuldade, decorrente da enorme flexibilidade de um modelo irrestrito de mistura, é, no entanto, evitada quando se adota o modelo denominado *Post-Nonlinear* (PNL) ou com não-linearidade posterior, que pode ser considerado híbrido por possuir uma seção linear seguida de uma seção não-linear [8]. Para esse modelo, a ICA volta a ser uma opção sólida e viável para a realização de BSS.

Para efetuar a separação de misturas PNL, adota-se tipicamente um sistema separador composto por um conjunto de não-linearidades monotônicas e por uma matriz inversível. Ademais, é preciso lançar mão, tipicamente, de uma função custo baseada numa métrica de informação mútua, a qual pode ser conduzir a múltiplos ótimos locais. Essas dificuldades evidenciam dois problemas fundamentais de projeto: a escolha cautelosa de modelos não-lineares que sejam inerentemente monotôni-

cos e a necessidade de técnicas capazes de realizar otimização em ambientes multimodais. Neste trabalho, é proposta uma metodologia que busca lidar com ambos os problemas, no âmbito de sinais reais, de uma forma tão ampla quanto possível. Para tanto, investigamos três estruturas monotônicas, todas com comprovada capacidade de aproximação:

- Polinômios com restrição de monotonicidade [9];
- Rede neural monotônica de Sill [10];
- Rede MLP Monotônica [11].

Além disso, propõe-se o uso de duas meta-heurísticas bio-inspiradas de otimização - que apresentam um interessante balanço entre complexidade de capacidade de busca global - para lidar com a tarefa de escolha dos parâmetros do sistema separador: o algoritmo CLONALG [12], um tipo de sistema imunológico artificial, e um algoritmo de enxame de partículas. Em todos os casos, a estimação da função custo adotada - a informação mútua - é feita por meio de uma metodologia baseada em estatísticas de ordem [13].

1.1 Organização

A presente dissertação está organizada da seguinte maneira:

- Capítulo 2: Neste capítulo, são apresentados os fundamentos da teoria de separação cega de fontes, juntamente com algumas aplicações e estratégias. Em seguida, introduzimos o esquema geral da ICA, bem como os principais critérios e algoritmos a ela associados. Finalmente, discute-se a extensão do problema de separação de fontes para o caso não-linear, com destaque para o modelo *Post-Nonlinear*.
- Capítulo 3: Neste capítulo, apresenta-se inicialmente a metodologia de estimação da informação mútua baseada em estatísticas de ordem. Posteriormente, são expostos os algoritmos bio-inspirados considerados neste trabalho - CLONALG e algoritmo de enxame de partículas - e as bases para sua aplicação junto ao problema de otimização associado ao modelo PNL.
- Capítulo 4: O capítulo é dedicado à descrição das estruturas monotônicas empregadas neste trabalho - polinômios com restrição de monotonicidade, rede neural de Sill e rede MLP monotônica - as quais permitem a construção de sistemas separadores de aplicabilidade geral.

- Capítulo 5: Esse capítulo contém os resultados experimentais que dão suporte ao trabalho. Primeiramente, realiza-se um estudo comparativo das meta-heurísticas de otimização e, em seguida, apresentam-se os ensaios relativos à abordagem geral proposta, que também abrange as estruturas monotônicas.
- Capítulo 6: Este trabalho se encerra com um capítulo contendo as conclusões gerais da dissertação, bem como perspectivas de trabalhos futuros nessa área.

Capítulo 2

Separação Cega de Fontes

Este capítulo é dedicado à apresentação dos conceitos da teoria de Separação Cega de Fontes (BSS, do inglês *Blind Source Separation*). Inicialmente, apresentaremos um breve histórico e um elenco de aplicações. Em seguida, descreveremos a caracterização matemática do problema, e, a partir desses conceitos, mostraremos como é possível obter métodos BSS através da hipótese de que as fontes são independentes entre si e de que possuem distribuição diferente da gaussiana. Esta abordagem, conhecida como Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*), é uma das principais ferramentas para solução do problema de BSS e será discutida no decorrer deste capítulo, tanto para sistemas lineares quanto para sistemas não-lineares. Em particular, discutiremos as dificuldades presentes no desenvolvimento de técnicas de ICA próprias para modelos de mistura com não-linearidade posterior.

2.1 Histórico

Os primeiros estudos sobre separação cega de fontes surgiram em meados dos anos 80, sendo o trabalho de Héroult, Jutten e Ans [14] considerado o marco inicial da área. Este trabalho teve origem a partir da ideia de estudar, num contexto de neurofisiologia, a estrutura empregada pelo sistema nervoso central para a ativação muscular. Em termos simples, o cérebro coordena todos os movimentos musculares, sejam eles amplos ou restritos. Para isso, ele utiliza um único sinal que codifica duas informações relevantes (no caso, deslocamento e velocidade angular do movimento). Uma vez que a tarefa de movimentação é geralmente realizada com sucesso, o objetivo de [14] era obter um método computacional capaz de distinguir essas duas informações. Uma das principais contribuições deste trabalho pioneiro foi mostrar a necessidade de utilizar estatísticas de ordem superior para lidar com essa separação. Além disso, esse mesmo trabalho colaborou para que a modelagem algébrica dos sistemas misturador e separador fosse introduzida.

Apesar de o trabalho [14] ter sido o marco inicial em BSS, somente no início da década de 1990 as pesquisas nessa área começaram a atrair maior atenção dos pesquisadores. Com os trabalhos de Pierre Comon e Jean-François Cardoso, o problema de separação cega de fontes foi formalizado matematicamente. Utilizando os trabalhos de Hérault *et al.* [14], a teoria de informação de Claude Shannon [15] e os resultados obtidos na década de 1950 por Darmois [16], Pierre Comon [7] elaborou um modelo matemático que descrevia a ideia da ICA, e esclareceu também de que forma a independência estatística se insere no problema de separação cega de fontes. Cardoso, por sua vez, entre outras coisas, conduziu estudos sobre o estimador de máxima verossimilhança em BSS [17], ele introduziu métodos tensoriais no problema [18] e participou do desenvolvimento da técnica do gradiente relativo [19].

Na mesma época em que os pesquisadores franceses apresentaram seus resultados, o trabalho de Bell e Sejnowski [20] deu suporte adicional à popularização da BSS. Os esforços desses pesquisadores, que lidaram com codificação neural e ICA, causaram grande impacto na comunidade de processamento de sinais, pois levaram a técnicas capazes de separar um considerável número de fontes de forma relativamente simples.

Também merecem destaque as contribuições de três pesquisadores finlandeses: Karhunen, Oja e Hyvärinen. Os trabalhos de Karhunen e Oja [21] se concentraram na ICA como extensão não-linear da técnica de Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*). Já Hyvärinen colaborou para o desenvolvimento de critérios baseados na maximização da não-gaussianidade [4], em que se baseia o FastICA, que é uma ferramenta muito utilizada em separação de fontes, por aliar robustez e simplicidade. O algoritmo, aliás, permitiu uma maior popularização de técnicas para resolução de problemas em BSS.

Até o fim da última década, a maioria das técnicas propostas haviam sido desenvolvidas para resolver o problema de BSS no contexto de misturas lineares e instantâneas. Porém, em alguns cenários práticos, como, por exemplo, quando devem ser usados certos sensores e amplificadores, é de grande importância lidar com modelos de mistura não-lineares. Devido a isso, tornou-se objeto de estudo o problema de BSS não-linear, e, desse esforço, emergiu o modelo de mistura com não-linearidade posterior (*post-nonlinear* - PNL) [8], que é particularmente atraente por permitir, sob certas condições, a recuperação das fontes através de ICA. Teremos a chance de discuti-lo melhor na Seção 2.7.

2.2 Aplicações

Os fundamentos presentes na formulação de BSS possibilita uma gama de aplicações bastante variada, compreendendo a inúmeros tipos de cenários. Veremos nesta seção algumas das mais impor-

tantes aplicações dessa área.

2.2.1 Separação de Sinais de Áudio - *Cocktail-party problem*

Um problema bem conhecido em separação cega de fontes é o *Cocktail Party Problem*, que surge, por exemplo, se várias pessoas estão em uma sala conversando ao mesmo tempo. O problema consiste basicamente em recuperar o sinal de voz de cada uma das pessoas a partir dos sinais misturados obtidos pelos sensores, como ilustrado pela Fig. 2.1.

Mesmo com toda essa miscelânea de informações, o ser humano, através de seu complexo e eficiente sistema nervoso central, é capaz de distinguir os sinais relevantes e separá-los, de forma a caracterizar o ambiente sonoro que está à sua volta. Esse processo é conhecido como *cocktail-party effect* [22], sendo que nosso cérebro é capaz de realizá-lo frequentemente e sem dificuldade. Essa capacidade mostrou a concepção de sistemas artificiais aptos a conduzir esse mesmo processo. Embora essa tarefa não seja trivial as técnicas de BSS via ICA são atualmente as mais indicadas para lidar com o problema [4].

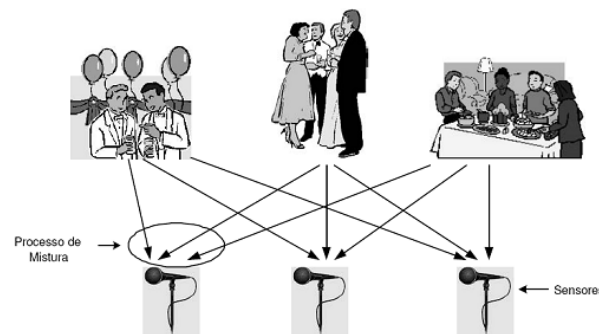


Figura 2.1: O *cocktail-party problem* (retirado de [2])

2.2.2 Separação de Imagens

Problemas de BSS também surgem quando se lida com imagens, o que ocorre por exemplo, em ultrassonografia [4] e processamento de imagens digitalizadas por *scanners* [23]. Um problema fundamental neste contexto é encontrar uma representação adequada de uma imagem que está misturada de alguma forma com uma outra.

Para exemplificar a tarefa de BSS apresentamos a seguir um exemplo simples. Na Fig. 2.2, tem-se as duas fontes originais, ou seja, os dados de entrada. Na Fig. 2.3, apresentamos duas misturas dessas fontes. O objetivo então é obter uma estimativa tão fiel quanto possível das duas imagens originais.



Figura 2.2: Imagens Originais



Figura 2.3: Imagens Misturadas

2.2.3 Telecomunicações - BSS e Equalização Cega de Canais

BSS também está relacionada a um tema de grande relevância em comunicações digitais: a equalização de canais [24]. Um sistema de comunicação é composto basicamente por três elementos: transmissor, canal e receptor. Idealmente, num sistema desse tipo, é necessário que a informação que chega ao receptor seja tão fiel quanto possível ao original. Para que isso ocorra, é preciso lançar mão de estratégias capazes de atenuar as distorções impostas pelo canal durante a transmissão do sinal. Uma possibilidade de grande interesse prático é a equalização de canal, em que se utiliza um filtro (equalizador) no receptor de modo que este seja capaz de inverter a ação do meio. O esquema básico da equalização é apresentado na Fig. 2.4. Neste caso, tem-se que $s(n)$ é o sinal transmitido, $x(n)$ é o sinal recebido e $y(n)$ é a estimativa do sinal transmitido.

O ponto crucial nesta aplicação é desenvolver técnicas que guiem o ajuste dos parâmetros livres do equalizador, levando a uma boa estimativa do sinal transmitido. Uma prática comum em comunicações é a utilização de períodos em que não é transmitida informação útil pelo canal, mas apenas uma sequência pré-definida de treinamento. Assim, com base no sinal degradado que chega ao receptor

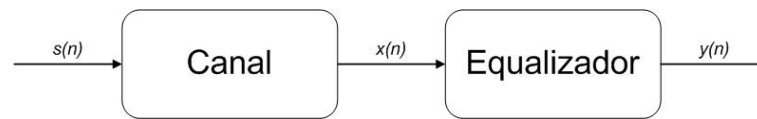


Figura 2.4: Esquema de Equalização

e no conhecimento prévio do que deveria ter sido recebido, ajusta-se o equalizador. Esta abordagem recebe o nome de *equalização supervisionada*.

Um paradigma alternativo leva à noção de equalização não-supervisionada (ou cega) [25]. Neste tipo de abordagem, tem-se o conhecimento apenas dos sinais recebidos e de algumas informações estatísticas dos sinais transmitidos, sem que seja necessário lançar mão de sequências pré-definidas.

Diante dessa breve descrição, percebemos que uma técnica de equalização cega busca recuperar o sinal transmitido através de um filtro no receptor, observando somente as amostras da saída do canal. Isso já insinua que equalização e BSS são similares. A diferença principal é que a equalização é definida classicamente em um cenário SISO (*Single-Input Single-Output*) e se baseia em filtragem temporal, enquanto a BSS aborda tipicamente sistemas MIMO (*Multiple-Input Multiple-Output*) e se baseia em filtragem espacial ou espaço-temporal. Porém, é perfeitamente possível formular o problema de equalização cega de canais SISO como uma tarefa de BSS [5] e [26].

2.2.4 Arranjo de Sensores Químicos

Inicialmente, esta aplicação foi estudada em [27]. Duarte, em seu trabalho de doutorado [6], abordou a tarefa de utilizar um arranjo de sensores químicos capaz de estimar as concentrações de íons presentes numa solução aquosa à luz de BSS não-linear. Nesse trabalho, foram utilizados eletrodos íon-seletivos, que são dispositivos capazes de medir concentrações iônicas específicas. O maior desafio encontrado neste trabalho foi o fato de que os eletrodos podem ser afetados pela interferência de outros íons que estão na solução. Portanto, o caminho seguido foi o uso de um arranjo de eletrodos seguido por bloco de processamento de sinal, projetado para extrair somente a informação relevante (concentração iônica) da solução.

Além da interferência, há outros desafios na utilização dessa técnica. Primeiro, o modelo de mistura associado ao arranjo de eletrodos é não-linear. Outro grande desafio é que nesta aplicação há tipicamente um número reduzido de amostras disponíveis, o que gera dificuldades para técnicas de separação que dependam de estimativas complexas.

2.3 Definição Matemática do Problema de BSS

Tendo descrito algumas aplicações relevantes ligadas ao problema de BSS, o próximo passo será conceituar matematicamente a tarefa de separação, mostrada na Fig. 2.5. De acordo com esta figura, as N fontes são representadas pelo vetor $\mathbf{s}(n) = [s_1(n) \ s_2(n) \ \dots \ s_N(n)]^T$ e os M sinais misturados são representados pelo vetor $\mathbf{x}(n) = [x_1(n) \ x_2(n) \ \dots \ x_M(n)]^T$.

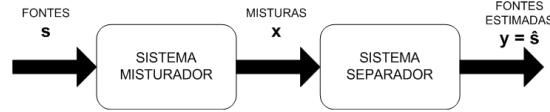


Figura 2.5: Diagrama Esquemático de Separação Cega de Fontes

A mistura se dá segundo algum mapeamento, que se relaciona com o vetor de fontes da seguinte forma:

$$\mathbf{x}(n) = \mathcal{F}(\mathbf{s}(n), \mathbf{s}(n-1), \dots, \mathbf{s}(n-L), \mathbf{r}(n)), \quad (2.1)$$

onde $\mathcal{F}(\cdot)$ corresponde ao mapeamento que descreve a ação do sistema misturador podendo ser não-linear, L corresponde ao número de atrasos presentes no processo de mistura e o vetor $\mathbf{r}(n)$ corresponde ao ruído, que pode estar contido tanto nas fontes quanto nos sensores.

A expressão (2.1) abrange portanto os seguintes efeitos: ruído, memória e não-linearidades, que, de um modo geral, estão presentes no problema de BSS. Porém não há técnicas de BSS capazes de lidar com todos esses efeitos num caso geral. Dessa forma, geralmente assume-se um modelo simplificado que representa casos particulares de (2.1). Sabendo disso, a seguir, será apresentado um esquema de classificação de sistemas misturadores.

2.3.1 Sistemas Lineares ou Não-Lineares

Um sistema misturador é dito linear se o mapeamento $\mathcal{F}(\cdot)$ atende ao princípio da superposição, ou seja, quando:

$$\mathcal{F}(a_1 \mathbf{s}_1(n) + a_2 \mathbf{s}_2(n)) = a_1 \mathcal{F}(\mathbf{s}_1(n)) + a_2 \mathcal{F}(\mathbf{s}_2(n)), \quad (2.2)$$

para quaisquer constantes a_1 e a_2 e vetores de sinais $\mathbf{s}_1(n)$ e $\mathbf{s}_2(n)$. Caso contrário, o sistema é dito não-linear.

2.3.2 Sistemas Instantâneos e com Memória

Caso o sistema apresente $L = 0$, ou seja, não haja dependência relativamente a atrasos, ele é dito instantâneo. Nas situações em que os sinais observados possuem representações de amostras atuais e passadas, ou seja, quando $L > 0$, o sistema é dito com memória, ou, no caso linear, convolutivo.

2.3.3 Relação entre Número de Fontes e Sensores

Quando o número de sensores é maior que o número de fontes ($M > N$), tem-se o chamado caso de mistura sobredeterminada. Em contrapartida, quando o número de sensores é inferior ao número de fontes ($M < N$), tem-se um sistema misturador subdeterminado.

O caso mais usual em BSS corresponde ao cenário com sistema misturador linear, instantâneo, e dotado do mesmo número de fontes e sensores, sem ruído. Esse modelo é caracterizado por uma matriz de mistura $\mathbf{A}_{N \times N}$, de maneira que o vetor de observações $\mathbf{x}(n)$, que são combinações lineares das fontes, seja expresso do seguinte modo na forma matricial (omitiremos o índice temporal por simplicidade):

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \quad (2.3)$$

No caso de (2.3), uma possibilidade natural é realizar a separação multiplicando à esquerda o vetor de mistura \mathbf{x} por uma matriz de separação \mathbf{W} , que, a princípio, deve ser $\mathbf{W} = \mathbf{A}^{-1}$:

$$\begin{aligned} \mathbf{y} &= \mathbf{W}\mathbf{x} \\ &= \mathbf{W}\mathbf{A}\mathbf{s}, \\ &= \mathbf{s}. \end{aligned} \quad (2.4)$$

Diante do problema de BSS que acabamos de apresentar, surge a seguinte questão: como podemos encontrar uma matriz \mathbf{W} sem qualquer conhecimento das amostras das fontes e dos elementos da matriz de mistura \mathbf{A} ?

Uma resposta para esta pergunta se relaciona com uma ferramenta chamada de Análise de Componentes Independentes (ICA), a qual exige a hipótese de que as fontes sejam estatisticamente independentes. Desta maneira, a ideia de ICA é separar as fontes a partir da busca de uma matriz \mathbf{W} que forneça um vetor \mathbf{y} cujos seus elementos também sejam estatisticamente independentes [4]. A ICA será apresentada na próxima seção.

2.4 Análise de Componentes Independentes

Como dito, uma abordagem clássica para resolver o problema de separação cega de fontes é a Análise de Componentes Independentes (ICA). De modo geral, a ICA é um método utilizado para encontrar fontes independentes e não-gaussianas a partir de um conjunto de dados multidimensional. A origem desse método está ligada ao problema de BSS, e é possível usá-lo tanto em cenários lineares quanto não-lineares. Em um primeiro momento, discutiremos o caso linear, que é o mais simples.

Por definição, a ICA de um vetor aleatório $\mathbf{x} = [x_1 x_2 \dots x_M]^T$ consiste na determinação de uma transformação linear $\mathbf{y} = \mathbf{W}\mathbf{x}$ de tal forma que os elementos do vetor aleatório $\mathbf{y} = [y_1 y_2 \dots y_N]^T$ sejam tão estatisticamente independentes quanto possível. Logo, o ponto fundamental da ICA é separar as fontes a partir da recuperação de independência.

Para a aplicação da ICA em BSS, além da restrição de que as fontes devam ser estatisticamente independentes, outro ponto importante se relaciona às suas funções de densidade de probabilidade: dentre todas as fontes, no máximo uma pode ser gaussiana [4] [28]. Isso se relaciona ao fato de que variáveis aleatórias gaussianas podem ser completamente caracterizadas através de estatísticas de segunda ordem [29]. No entanto, como veremos na Seção 2.4.1, essas estatísticas não são suficientes para realizar a separação das fontes.

2.4.1 Independência Estatística x Descorrelação

O conceito de independência estatística é central em teoria de probabilidade, além de ser fundamental para a análise de componentes independentes. Matematicamente, a independência estatística é definida em termos das densidades de probabilidade. Mais precisamente, se a densidade conjunta de variáveis aleatórias puder ser fatorada em um produto de densidades marginais, isto é,

$$p_{1,2,\dots,n}(y_1, y_2, \dots, y_n) = p_1(y_1)p_2(y_2) \dots p_n(y_n), \quad (2.5)$$

então tais variáveis são estatisticamente independentes.

Tendo apresentado o conceito de independência estatística, é importante também estabelecer o conceito de decorrelação, que é menos restritivo. Matematicamente, diz-se que um conjunto de n variáveis aleatórias é formado por variáveis decorrelacionadas se suas covariâncias par a par são nulas, como definido pela Equação (2.6):

$$E\{x_i x_j\} = E\{x_i\}E\{x_j\} \quad \forall i, j, \quad (2.6)$$

Com base em (2.5) e (2.6), pode-se mostrar que, se as variáveis aleatórias são estatisticamente independentes, elas são decorrelacionadas, mas, por outro lado, a decorrelação não implica em

independência. O único caso em que isso não se aplica ocorre quando as variáveis aleatórias são gaussianas, uma vez que estas variáveis são completamente descritas por estatísticas de segunda ordem [30]. Como a decorrelação em geral não garante a independência, ela também não pode garantir a separação em componentes independentes. Assim, para assegurar a separação é necessário o uso de estatísticas de ordem superior, o que faz com que a separação o emprego da ICA somente seja adequado se houver no máximo uma fonte gaussiana.

De acordo com o teorema de Darmois [7], é possível separar as fontes com base na recuperação da independência estatística, desde que as fontes e o sistema separador satisfaçam algumas condições. Para entendermos essa questão, é importante descrever em mais detalhes o conceito de separabilidade.

2.4.2 Separabilidade

Para definir a separabilidade através da ICA, retornaremos ao modelo de mistura (2.3). Nesse modelo as estimativas das fontes são dadas pela equação $\mathbf{y} = \mathbf{W}\mathbf{x}$. O modelo de mistura (2.3) é *separável* caso a matriz de separação \mathbf{W} que leve a um vetor \mathbf{y} cujos elementos são estatisticamente independentes entre si implique $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{D}\mathbf{P}\mathbf{s}$, onde \mathbf{D} corresponde uma matriz diagonal e \mathbf{P} a uma matriz de permutação. Deste modo é possível recuperar as fontes através da ICA. Note que a definição de um sistema *separável*, suscita duas ambiguidades:

1. Não é possível determinar a variância dos componentes independentes.
2. Não é possível determinar a ordem de recuperação dos componentes independentes.

Tais ambiguidades podem parecer uma limitação da técnica de separação, mas decorrem do fato de que a independência estatística entre os elementos de um vetor não é alterada por permutações e fatores de escala. Isso não acarreta necessariamente dificuldades, pois, em muitos casos a informação desejada se encontra nas formas de onda recuperadas.

As condições para que as fontes sejam recuperadas podem ser resumidas através do seguinte teorema [7]:

Teorema 1 (Separação em ICA). *O modelo $\mathbf{x} = \mathbf{A}\mathbf{s}$ é separável se e somente se a matriz \mathbf{A} possuir posto completo, e, no máximo, um dos elementos do vetor aleatório \mathbf{s} for gaussiano.*

Até agora, vimos os conceitos que envolvem o problema de separação de fontes e a ferramenta ICA, tipicamente utilizada para recuperar essas fontes. Porém, a aplicabilidade de ICA em BSS depende da escolha de critérios e estratégias adequados. A seguir, discutiremos algumas escolhas representativas para se realizar separação através de ICA.

2.5 Análise de Componentes Independentes - ICA: Critérios e Estratégias

Nesta seção, descreveremos as principais estratégias desenvolvidas para ICA no contexto mais didático de misturas lineares, uma vez que tais técnicas são fundamentais para a solução dos problemas a serem tratados nos próximos capítulos.

2.5.1 Informação Mútua

O critério de informação mútua permite a quantificação do grau de dependência estatística entre variáveis aleatórias. Diante disso, tal critério é um importante candidato a função custo para o problema de BSS.

A ideia de informação mútua se vincula à definição de entropia. Deste modo, antes de formalizá-la, é necessário definir esta última grandeza. Classicamente, no contexto da teoria da informação [31], a entropia é compreendida como uma medida de incerteza de uma variável aleatória. Matematicamente, pode-se definir, para um vetor de variáveis aleatórias contínuas \mathbf{x} , sua entropia diferencial¹ $H(\mathbf{x})$ como:

$$H(\mathbf{x}) = -E\{\log[p_{\mathbf{x}}(\mathbf{x})]\} = - \int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \log(p_{\mathbf{x}}(\mathbf{x})) d\mathbf{x}. \quad (2.7)$$

De maneira similar, pode-se definir a entropia condicional, que mede a incerteza de um vetor de variáveis aleatórias \mathbf{x} após a observação de um outro vetor \mathbf{y} . Matematicamente, a entropia condicional de \mathbf{x} dado \mathbf{y} é:

$$H(\mathbf{x} | \mathbf{y}) = -E\{\log[p_{\mathbf{x}|\mathbf{y}}(\mathbf{x} | \mathbf{y})]\} = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) \log(p_{\mathbf{x}|\mathbf{y}}(\mathbf{x} | \mathbf{y})) d\mathbf{x} d\mathbf{y}, \quad (2.8)$$

onde:

$$p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) = p_{\mathbf{x}|\mathbf{y}}(\mathbf{x} | \mathbf{y}) p_{\mathbf{y}}(\mathbf{y}), \quad (2.9)$$

é a função densidade de probabilidade conjunta de \mathbf{x} e \mathbf{y} .

Assim, pode-se definir a informação mútua entre duas variáveis aleatórias vetoriais \mathbf{x} e \mathbf{y} como:

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{x}) - H(\mathbf{x} | \mathbf{y}). \quad (2.10)$$

¹A entropia também pode ser definida para variáveis discretas, embora esse caso não seja relevante para este trabalho. No contexto discreto, $H(\mathbf{x}) = -\sum_i p_i \log p_i$.

A informação mútua é não-negativa e assume valor zero se e somente se \mathbf{x} e \mathbf{y} forem independentes. É por isso que, como dito anteriormente, o critério de informação mútua é um forte candidato a compor uma função custo na utilização da ICA. Além disso, a informação mútua pode também ser definida entre componentes de um único vetor de variáveis aleatórias \mathbf{y} :

$$I(\mathbf{y}) = -H(\mathbf{y}) + \sum_{i=1}^N H(y_i). \quad (2.11)$$

A expressão (2.11) indica que minimizar a informação mútua equivale a tornar a entropia conjunta do vetor \mathbf{y} a mais próxima possível da soma das entropias marginais [28].

Uma maneira alternativa para definir a informação mútua surge da divergência de *Kullback-Leibler* [7]. Essa grandeza, embora rigorosamente não seja uma distância, pois, por exemplo, não é simétrica, pode ser usada como uma medida de similaridade entre duas densidades de probabilidade. De fato, ela nunca será negativa e será nula se e somente se as densidades forem iguais. A divergência de *Kullback-Leibler* entre duas funções multidimensionais $f(\mathbf{r})$ e $g(\mathbf{r})$ é dada por:

$$D(f(\mathbf{r}), g(\mathbf{r})) = \int f(\mathbf{r}) \log \left(\frac{f(\mathbf{r})}{g(\mathbf{r})} \right) d\mathbf{r}. \quad (2.12)$$

A informação mútua surge quando uma das funções corresponde à densidade de probabilidade conjunta de um vetor aleatório \mathbf{y} e a outra denota o produto das densidades marginais dos elementos deste vetor. Matematicamente, esse caso é expresso do seguinte modo:

$$\begin{aligned} I(\mathbf{y}) &= \int p_{\mathbf{y}}(\mathbf{y}) \log \left(\frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^N p_{y_i}(y_i)} \right) d\mathbf{y} \\ &= D(p_{\mathbf{y}}(\mathbf{y}) \parallel p_{y_1}(y_1) p_{y_2}(y_2) \dots p_{y_N}(y_N)). \end{aligned} \quad (2.13)$$

2.5.2 Estimação por Máxima Verossimilhança

Uma abordagem muito popular em teoria de estimação é a *estimação por máxima verossimilhança* [32]. Nesta seção, descreveremos como essa consagrada técnica pode ser aplicada para resolver o problema de separação. Porém, antes, é importante revisar alguns conceitos fundamentais.

O problema de estimação de parâmetros a determinar um estimador para os parâmetros $\theta = [\theta_1, \dots, \theta_p]$ a partir de um conjunto de amostras $\mathbf{e} = [e(1), \dots, e(J)]$. Na abordagem por máxima verossimilhança, as estimativas de θ , que são denotadas por $\tilde{\theta}$, são alcançadas através da maximização da função de verossimilhança $L(\theta)$. Matematicamente, temos que:

$$\tilde{\theta} = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} p_{\mathbf{e}}(\mathbf{e} \mid \theta), \quad (2.14)$$

ou seja, a metodologia de máxima verossimilhança procura um conjunto de parâmetros que maximize a probabilidade condicional de \mathbf{e} dado θ .

Considerando os elementos de $\mathbf{e} = [e(1), \dots, e(J)]$ estatisticamente independentes, a função de verossimilhança é dada por:

$$L(\theta) = p_{\mathbf{e}}(\mathbf{e} | \theta) = \prod_{j=1}^J p_e(e(j) | \theta). \quad (2.15)$$

No problema de BSS, o emprego da estimação por máxima verossimilhança se dá para determinação dos elementos da matriz de separação \mathbf{W} . Sabendo que os dados disponíveis são as amostras das misturas, representadas por $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(J)]$, podemos escrever:

$$L(\mathbf{W}) = \prod_{j=1}^J p_{\mathbf{x}}(\mathbf{x}(j) | \mathbf{W}). \quad (2.16)$$

Tendo em vista que vale a seguinte relação [29]:

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{p_s(\mathbf{A}^{-1}\mathbf{x})}{|\det(\mathbf{A})|} = p_s(\mathbf{W}\mathbf{x}) |\det(\mathbf{W})|, \quad (2.17)$$

onde \mathbf{A} é a matriz do sistema misturador, e que as fontes são independentes, podemos escrever a função de verossimilhança, em termos da matriz \mathbf{W} , da seguinte forma:

$$L(\mathbf{W}) = \prod_{j=1}^J \prod_{n=1}^N p_{s_n}(\mathbf{w}_n \mathbf{x}(j)) |\det(\mathbf{W})|, \quad (2.18)$$

onde \mathbf{w}_n denota a n -ésima linha da matriz \mathbf{W} .

Com a função de verossimilhança em termos da matriz \mathbf{W} obtida em (2.18), podemos fazer duas considerações simplificadoras. Uma delas é transformar os produtórios em somatórios utilizando uma versão logarítmica. A outra é lançar mão da lei dos grande números, que foi utilizada em [17], para obter uma variante probabilística.

A primeira ideia de simplificação nos leva a:

$$\frac{1}{J} \log(L(\mathbf{W})) = \frac{1}{J} \sum_{j=1}^J \sum_{n=1}^N \log(p_{s_n}(\mathbf{w}_n \mathbf{x}(j))) + \log(|\det(\mathbf{W})|). \quad (2.19)$$

Considerando a lei dos grandes números, tem-se que:

$$\frac{1}{J} \log(L(\mathbf{W})) = E\left\{ \sum_{n=1}^N \log(p_{s_n}(\mathbf{w}_n \mathbf{x}(j))) \right\} + \log(|\det(\mathbf{W})|) \quad (2.20)$$

A partir da expressão (2.20), a abordagem por máxima verossimilhança em BSS pode ser formulada em termos da divergência de *Kullback-Leibler*, como foi demonstrado em [28]:

$$\widetilde{\mathbf{W}} = \arg \max_{\mathbf{W}} (L(\mathbf{W})) \triangleq \arg \max_{\mathbf{W}} (D(p_{\mathbf{W}\mathbf{x}}(\mathbf{r})) \mid p_s(\mathbf{r})) \quad (2.21)$$

Com esses conceitos, podemos concluir que a estimação por máxima verossimilhança baseia-se na escolha de uma matriz \mathbf{W} que minimize a divergência de *Kullback-Leibler* entre as funções de densidade de probabilidade das fontes \mathbf{s} e de suas estimativas \mathbf{y} . No entanto, esta estratégia exige o conhecimento das pdf's das fontes, o que pode ser inviável num caso estritamente não-supervisionado.

2.5.3 Infomax

Um outro critério muito utilizado em separação cega de fontes é o *Infomax* (*Information Maximization*), que se vincula ao uso de redes neurais artificiais com treinamento não-supervisionado baseado em elementos da teoria da informação. Inicialmente, este critério foi desenvolvido por Linsker [33], mas, no contexto de separação cega de fontes, a abordagem foi apresentada por Bell e Sejnowski [20]. A Fig. 2.6 ilustra o modelo de separação introduzido por esses autores.

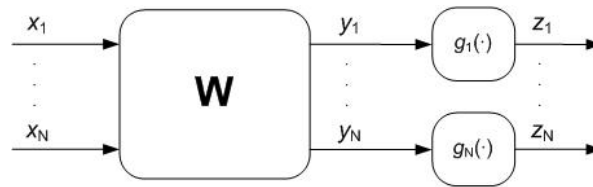


Figura 2.6: Estrutura do sistema separador no critério Infomax (retirado de [2])

De acordo com Bell e Sejnowski, o princípio Infomax leva a uma estrutura em dois estágios: um estágio linear, que é representado pela matriz \mathbf{W} , seguido de um estágio não-linear, que é composto por funções não-lineares monotonicamente crescentes ², com $g_i(-\infty) = 0$ e $g_i(\infty) = 1$. Estas funções são representadas por: $\mathbf{g}(\cdot) = [g_1(\cdot) \dots g_N(\cdot)]$. Matematicamente, a estrutura da Fig. 2.6 pode ser representada por:

$$\mathbf{z} = \mathbf{g}(\mathbf{y}) = \mathbf{g}(\mathbf{W}\mathbf{x}) = [g_1(\mathbf{w}_1\mathbf{x}) \dots g_N(\mathbf{w}_N\mathbf{x})]^T \quad (2.22)$$

Um fato que devemos salientar nessa estrutura da rede neural é que as estimativas das fontes são fornecidas pelas saídas do estágio linear. Porém, o estágio não-linear possui importância no ajuste dos parâmetros da rede. No critério Infomax, os coeficientes da rede neural devem ser ajustados de

²O leitor perceberá que a estrutura Infomax possui similaridade com a arquitetura de um separador PNL, que será visto na Seção 2.7, embora as não-linearidades tenham papel distinto em ambos os casos.

forma a *maximizar a transferência de informação* entre as entradas e saídas. Em outras palavras, busca-se adaptar a matriz \mathbf{W} de modo a *maximizar a informação mútua* entre \mathbf{x} e \mathbf{z} .

Em termos matemáticos, o critério, Infomax pode ser formulado através do seguinte problema de otimização:

$$\max_{\mathbf{W}} I(\mathbf{z}, \mathbf{x}) = H(\mathbf{z}) - H(\mathbf{z} | \mathbf{x}), \quad (2.23)$$

onde $I(\mathbf{z}, \mathbf{x})$ corresponde à informação mútua entre \mathbf{z} e \mathbf{x} .

Dado que o mapeamento de entrada-saída em (2.22) é determinístico e inversível, a entropia condicional $H(\mathbf{z} | \mathbf{x})$ não depende de \mathbf{W} . Logo, o critério Infomax é equivalente à maximização da entropia das saídas dessa rede, ou seja, pode-se escrever:

$$\max_{\mathbf{W}} I(\mathbf{z}, \mathbf{x}) = H(\mathbf{z}). \quad (2.24)$$

Ainda segundo este mesmo raciocínio, podemos mostrar que a entropia conjunta das saídas pode ser expressa da seguinte maneira:

$$H(\mathbf{z}) = H(\mathbf{x}) + E\left\{\sum_{i=1}^N \log(g'_i(\mathbf{w}_i \mathbf{x}))\right\} + \log(|\det(\mathbf{W})|), \quad (2.25)$$

onde $g'_i(\cdot)$ representa a derivada primeira da função $g_i(\cdot)$. Como a entropia conjunta dos dados de \mathbf{x} não depende da matriz de separação, esta não influenciará no processo de adaptação, e, portanto, a aplicação do Infomax no problema de BSS linear resulta na maximização da seguinte função custo:

$$J_{infomax}(\mathbf{W}) \triangleq E\left\{\sum_{i=1}^N \log(g'_i(\mathbf{w}_i \mathbf{x}))\right\} + \log(|\det(\mathbf{W})|). \quad (2.26)$$

Note que essa formulação possui certa similaridade com a estimação por máxima verossimilhança (2.20), o que, aliás, foi observado por Cardoso [32].

No item seguinte, apresentaremos o algoritmo correspondente ao critério Infomax, cuja finalidade é o ajuste adaptativo do sistema separador.

Algoritmo Infomax

Na literatura, o algoritmo que implementa a ideia do Infomax é conhecido como algoritmo de Bell-Sejnowski [20]. Esta técnica propõe uma otimização de (2.26) baseada no método do gradiente. Neste caso, o gradiente com relação à matriz \mathbf{W} é dado por:

$$\frac{\partial J_{\text{infomax}}(\mathbf{W})}{\partial \mathbf{W}} = E\{\mathbf{f}(\mathbf{W}\mathbf{x})\mathbf{x}^T\} + (\mathbf{W}^T)^{-1}, \quad (2.27)$$

onde $\mathbf{f}(\cdot) = [f_1(\cdot) \dots f_N(\cdot)]$ é um vetor de funções tais que $f_i(x) = d\log(g'_i(x))/dx$. Assim, é possível escrever a seguinte regra de atualização:

$$\mathbf{W} \leftarrow \mathbf{W} + \mu\{E\{\mathbf{f}(\mathbf{W}\mathbf{x})\mathbf{x}^T\} + (\mathbf{W}^T)^{-1}\}, \quad (2.28)$$

onde μ corresponde ao passo de adaptação. Uma versão estocástica deste algoritmo pode ser obtida omitindo o operador de esperança na expressão (2.28). O algoritmo mostrado pode convergir lentamente dado que a inversão da matriz \mathbf{W} acontece a cada passo. Para contornar esses problemas, pode-se utilizar o branqueamento dos dados e, especialmente, o gradiente natural [4].

2.5.4 Maximização da Não-Gaussianidade - Negentropia, Curtose e FastICA

Para entendermos como a maximização da não-gaussianidade se insere no problema de BSS/ICA, iniciaremos este tópico descrevendo o teorema central do limite, que é um resultado clássico da teoria da probabilidade. Segundo este teorema, a função densidade de probabilidade da soma de variáveis aleatórias independentes, sob certas condições, tende a uma variável gaussiana [29]. Deste modo, temos que a pdf da soma de duas variáveis aleatórias independentes está mais "próxima" de uma densidade gaussiana do que qualquer uma das variáveis originais.

Diante do teorema central do limite, podemos concluir que as misturas são mais semelhantes a gaussianas do que as fontes. Desta observação, surge a possibilidade de ajustar \mathbf{W} de forma a maximizar a não-gaussianidade das estimativas das fontes. Para isso, são utilizadas tipicamente duas medidas, curtose e negentropia, que quantificam a proximidade de uma variável aleatória qualquer em relação a uma variável gaussiana.

Em ICA como acabamos de expor, o uso dessas medidas se fundamenta na ideia de que as estimativas das fontes devem ser tão distantes quanto possível de uma variável gaussiana. Uma característica importante desse procedimento é a possibilidade de recuperar cada uma das fontes individualmente. Além disso, medidas de não-gaussianidade são utilizadas no algoritmo mais popular em BSS, o FastICA [4], que será descrito mais adiante.

A curtose, um cumulante de quarta ordem, é uma medida clássica de não-gaussianidade [29]. A curtose de uma variável aleatória real y de média zero é definida como:

$$\text{Kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2. \quad (2.29)$$

Uma interessante propriedade da curtose é que esta medida é não-nula para a grande maioria das

variáveis aleatórias, sendo a gaussiana uma das poucas exceções à regra. Deste modo, um critério de maximização da não-gaussianidade pode ser obtido através da maximização do valor absoluto desta grandeza para cada uma das estimativas individualmente, caso se queira determinar apenas algumas fontes, ou conjuntamente, caso queira estimar todas as fontes. Curtoses negativas são chamadas de subgaussianas. Curtoses positivas são chamadas de supergaussianas.

Outra opção de medida de não-gaussianidade, estabelecida em um contexto mais associado à teoria de informação, é a negentropia, definida como:

$$J_{Negentropia}(y) = H(y_{gauss}) - H(y), \quad (2.30)$$

onde y_{gauss} corresponde a uma variável aleatória gaussiana com a mesma variância que y . Esta medida é, de certo modo, mais robusta em relação *outliers* se comparada à curtose. Como vimos na Seção 2.5.1, a entropia de uma variável aleatória está relacionada à incerteza média dessa variável. Uma variável gaussiana tem a maior entropia dentre todas as variáveis aleatórias de mesma média e variância [31]. Portanto, a negentropia corresponde a uma medida de não-gaussianidade, já que se anula para uma variável gaussiana e é não negativa.

Uma dificuldade de se aplicar a negentropia diretamente no problema de BSS é a necessidade de estimar as entropias. Mas, de acordo com [34], é possível obter uma boa estimativa da negentropia utilizando a seguinte aproximação baseada em funções não-lineares:

$$J(y) = \alpha(E\{G(y)\} - E\{G(v)\})^2, \quad (2.31)$$

onde $G(\cdot)$ é uma função não-linear e não-quadrática, α é uma constante e v é uma variável aleatória gaussiana de mesma média e variância que y . Um dado importante observado em [4] é que, escolhendo a função $G(\cdot)$ de forma que ela não cresça muito rapidamente, obtêm-se estimadores mais robustos e computacionalmente simples.

2.5.5 FastICA

O FastICA, um algoritmo muito popular na área de separação cega de fontes, se baseia na maximização da não-gaussianidade. Para exemplificar a utilização deste algoritmo, iremos considerar a maximização da negentropia aproximada, expressa em (2.31), embora ele também possa ser utilizado na maximização da curtose. Além disso, iremos considerar a recuperação de uma fonte, ou seja, ajustaremos uma das linhas da matriz \mathbf{W} , denotada por \mathbf{w}_i^T , de modo que $y_i = \mathbf{w}_i^T \mathbf{x}$ resulte em uma boa estimativa da mesma. Podemos descrever esse problema de otimização da seguinte maneira:

$$\tilde{\mathbf{w}}_i = \arg \max_{\mathbf{w}_i} (E\{G(y_i)\} - E\{G(v)\})^2, \quad (2.32)$$

com restrição que $E\{y_i^2\} = E\{(\mathbf{w}_i^T \mathbf{x})^2\} = 1$. Através do método de Lagrange, é possível verificar que o problema de otimização é resolvido quando vale a seguinte condição:

$$E\{\mathbf{x}G'(\mathbf{w}_i^T \mathbf{x})\} + \beta \tilde{\mathbf{w}}_i = 0, \quad (2.33)$$

onde β é uma constante que depende do valor ótimo de $\tilde{\mathbf{w}}_i$.

Levando em conta algumas aproximações referentes ao fato de que os sinais que compõe a mistura já estão branqueados³, pode-se utilizar o método de Newton para resolver a equação (2.33), obtendo-se a seguinte regra de atualização:

$$\begin{aligned} \mathbf{w}_i &\leftarrow E\{\mathbf{x}G'(\mathbf{w}_i^T \mathbf{x})\} - E\{\mathbf{x}G''(\mathbf{w}_i^T \mathbf{x})\}\mathbf{w}_i \\ \mathbf{w}_i &\leftarrow \mathbf{w}_i / \|\mathbf{w}_i\|. \end{aligned} \quad (2.34)$$

Uma função não-linear normalmente empregada para $G'(y)$ é a tangente hiperbólica [34].

A partir da regra de atualização (2.34), é possível determinar as N fontes. Para que isso aconteça, é necessário executar a otimização para os N vetores \mathbf{w}_i . Porém, é necessário verificar e controlar as diversas execuções do algoritmo, para que não seja estimada sempre a mesma fonte. Uma maneira de resolver esse problema é utilizar a informação de que o sistema separador pode ser representado por uma matriz ortogonal, já que os sinais misturados são supostos descorrelacionados. Para isso, basta inserir no algoritmo alguma restrição que garanta a ortogonalidade da matriz \mathbf{W} , como pode ser visto no passo 4 da Tab. 2.1, que descreve o *modus operandi* do FastICA.

³O conceito de branqueamento será visto na próxima seção.

Tabela 2.1: Algoritmo FastICA utilizando a Negentropia como critério de otimização

1. Pré-processamento dos dados: centralizar os dados a fim de que eles tenham média nula e branqueá-los. Os dados resultantes formam os vetores \mathbf{x} ;
2. Inicializar os vetores \mathbf{w}_i aleatoriamente, mas obedecendo à restrição de norma unitária;
3. Para todo $i = 1, 2, \dots, N$, sendo N o número total de fontes,

$$\begin{aligned}\mathbf{w}_i &\leftarrow E\{\mathbf{x}G'(\mathbf{w}_i^T \mathbf{x})\} - E\{G''(\mathbf{w}_i^T \mathbf{x})\}\mathbf{w}_i \\ \mathbf{w}_i &\leftarrow \mathbf{w}_i / \|\mathbf{w}_i\|\end{aligned}$$

4. Organizar os vetores \mathbf{w}_i em uma matriz $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N)^T$ e realizar o processo de ortogonalização simétrica;

$$\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}\mathbf{W}$$

5. Se o algoritmo não convergir, retornar para o passo 3.

Resumindo, as principais estratégias utilizadas em ICA, vistas na Seção 2.4, podem ser visualizadas conforme a Fig. 2.7.

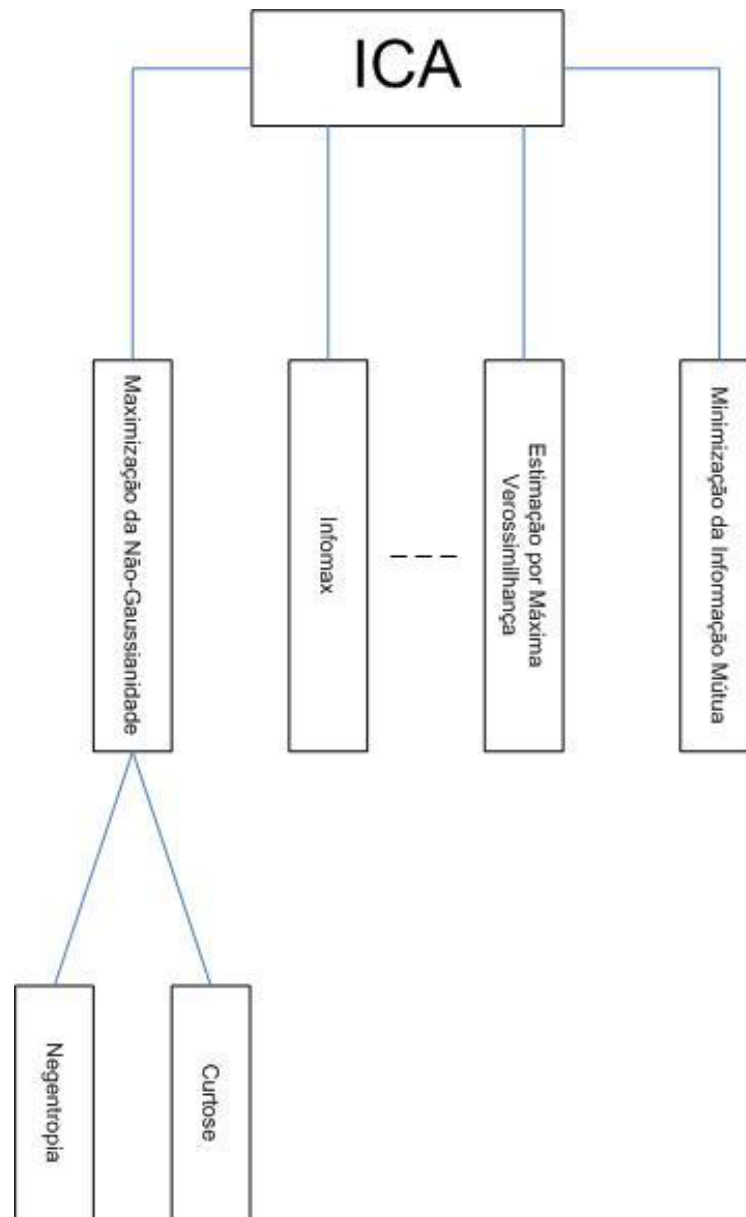


Figura 2.7: Estratégias para ICA

2.5.6 Análise por Componentes Principais (PCA), Branqueamento e PCA Não-Linear

A Análise por Componentes Principais (PCA, do inglês *Principal Component Analysis*) é uma técnica clássica em análise estatística, que pode ser aplicada em tarefas de extração de características e compressão de dados [4]. A ideia subjacente a essa técnica é encontrar projeções lineares que levem à representação dos dados com baixa redundância, gerando dessa forma uma representação

parcimoniosa. No método de PCA, faz-se uso apenas da informação trazida pela matriz de correlação, em contraste com ICA, que, como vimos, utiliza estatísticas de ordem superior a dois.

Antes de se realizar a PCA, os dados são centralizados para que sua média seja forçosamente nula. O número de componentes principais, N , é menor ou igual à dimensão M dos vetores de dados \mathbf{x} . Para encontrar os componentes principais que correspondem a um vetor \mathbf{y} , é feita a busca por vetores \mathbf{w}_i de forma a encontrar projeções de máxima variância. Como a variância de saída depende tanto da norma quanto da orientação do vetor \mathbf{w}_i , e cresce conforme a norma cresce, na prática, é imposta a restrição de que a norma desse vetor seja mantida em 1. Em outras palavras, busca-se, a princípio, um vetor $\mathbf{w}_1 = [w_{11} \dots w_{M1}]^T$ de modo que a seguinte combinação linear possua máxima variância:

$$y_1 = \mathbf{w}_1^T \mathbf{x}. \quad (2.35)$$

Nesta situação, diz-se que y_1 é a primeira componente principal de \mathbf{x} . A segunda componente principal y_2 é determinada de maneira semelhante, mas, como estamos tratando de um caso de eliminação de redundância, não deve existir correlação entre y_1 e y_2 . A expressão para y_2 é:

$$y_2 = \mathbf{w}_2^T \mathbf{x}. \quad (2.36)$$

Para garantir que a correlação entre y_1 e y_2 seja nula, é necessário que $\mathbf{w}_1^T \mathbf{w}_2 = 0$. Generalizando a ideia, a i -ésima componente principal é dada por:

$$y_i = \mathbf{w}_i^T \mathbf{x}, \quad (2.37)$$

com $\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij}$, para todo j , onde δ_{ij} corresponde à função delta de Kronecker.

Um outro caminho para expor o problema de PCA é a partir da minimização do erro quadrático médio (MMSE, do inglês *Minimum Mean Square Error*). Nesse contexto, a determinação dos vetores base \mathbf{w}_i é feita através da minimização da seguinte função custo:

$$J_{PCA} = E\{\|\mathbf{x} - \sum_{i=1}^N (\mathbf{w}_i^T \mathbf{x}) \mathbf{w}_i\|^2\}, \quad (2.38)$$

com $\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij}$, conforme explicado anteriormente. Deve-se observar que, quanto maior for o número de componentes principais na compressão, menor será o erro associado ao processo.

É possível mostrar que a solução que minimiza a equação (2.38) está relacionada com os autovetores da matriz de correlação $\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\}$, lembrando que supomos que os elementos de \mathbf{x} possuem média nula. Os autovalores dessa matriz fornecem os subsídios necessários à ordenação das componentes principais, sendo que o autovetor associado com o maior autovalor gera o primeiro deles. Vale ressaltar que a solução de PCA baseada diretamente na decomposição em autovalores e autovetores

pode não ser viável em diversas aplicações em tempo real. Diante disso, técnicas adaptativas foram desenvolvidas, dentre as quais podemos citar redes neurais artificiais [35].

Branqueamento

Um importante aspecto da utilização de análise por componentes principais (PCA) diz respeito ao caso em que $N = M$. Em tal situação, não há compressão de dados, sendo que a transformação resulta em um vetor aleatório descorrelacionado. Este procedimento é chamado de branqueamento espacial, sendo, em muitos casos, o primeiro passo para se realizar a separação cega de fontes.

Com o intuito de demonstrar o processo de branqueamento no problema de BSS, tomemos como base o modelo de mistura (2.3). Diante deste modelo, podemos obter a matriz de correlação entre as misturas, assumindo, como de praxe, que o vetor \mathbf{x} tem média nula:

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{A}\mathbf{R}_s\mathbf{A}^T = \mathbf{A}\mathbf{A}^T, \quad (2.39)$$

onde \mathbf{R}_s corresponde à matriz de correlação das fontes.

Em separação cega de fontes, temos que as fontes estimadas são dadas pela equação:

$$\mathbf{y} = \mathbf{W}\mathbf{x}.$$

A matriz de correlação após o sistema separador é dada por:

$$\mathbf{R}_y = \mathbf{W}\mathbf{R}_x\mathbf{W}^T. \quad (2.40)$$

Para branquear as saídas, é necessário determinar \mathbf{W} , de modo que:

$$\mathbf{R}_y = \mathbf{I} \rightarrow \mathbf{W}\mathbf{R}_x\mathbf{W}^T = \mathbf{I}, \quad (2.41)$$

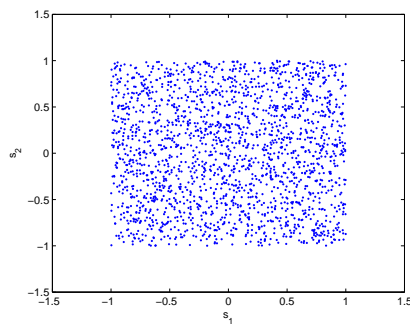
onde \mathbf{I} é a matriz identidade.

Uma solução para determinar \mathbf{W} é fazer o uso do método da decomposição em autovalores e autovetores da matriz de correlação das misturas, $\mathbf{R}_x = \mathbf{E}\mathbf{D}\mathbf{E}^T$, sendo \mathbf{E} uma matriz ortogonal cujas colunas representam os autovetores de norma unitária de \mathbf{R}_x e \mathbf{D} é uma matriz diagonal contendo os autovalores de \mathbf{R}_x . A matriz \mathbf{W} pode então ser computada da seguinte maneira:

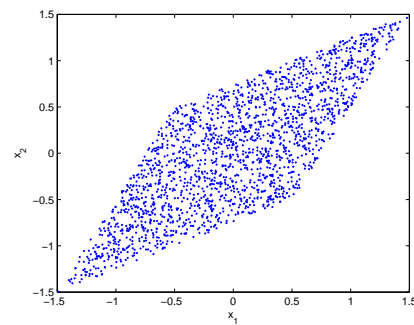
$$\mathbf{W} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T. \quad (2.42)$$

Note que a matriz \mathbf{W} garante apenas que as estimativas de \mathbf{y} sejam descorrelacionadas, e não necessariamente independentes. Para ilustrar as implicações desse fato, tomemos um exemplo em que cabe

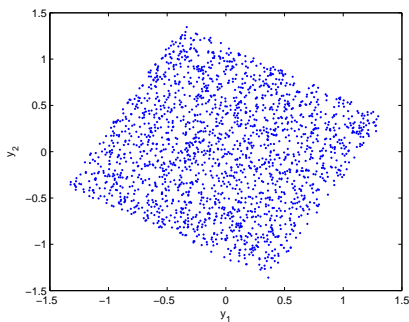
a essa matriz efetuar a separação, ilustrado na Fig. 2.8. No caso, temos na Fig. 2.8(a) a distribuição conjunta das fontes, ao passo que a Fig. 2.8(b) mostra a distribuição das misturas. Na Fig. 2.8(c) observamos a distribuição das estimativas obtidas via branqueamento. Como se assume que o sistema misturador é linear, as misturas são geradas a partir de escalonamentos e rotações das fontes. Apesar de a recuperação via branqueamento fornecer estimativas decorrelacionadas, a efetiva separação ainda exigiria uma operação de rotação.



(a) Distribuição conjunta das fontes.



(b) Distribuição conjunta das misturas.



(c) Distribuição conjunta das estimativas obtidas a partir do branqueamento das misturas.

Figura 2.8: Tratamento da BSS considerando estatística de segunda ordem

PCA Não-Linear

A abordagem de PCA linear possui, como vimos, a significativa limitação de não recuperar totalmente as fontes. Tal limitação, no entanto, pode ser contornada através da introdução de não-linearidades. Para isso, emprega-se a Análise por Componentes Principais Não-Linear (NPCA, do inglês - *Nonlinear Principal Component Analysis*) [4].

Utilizando o mesmo critério de minimização do erro quadrático médio visto anteriormente, a estimação da matriz \mathbf{W} através da ideia de NPCA é dada pela minimização do seguinte problema de otimização:

$$J_{NPCA} = E\{\|\mathbf{x} - \sum_{i=1}^N (g_i(\mathbf{w}_i^T \mathbf{x})) \mathbf{w}_i\|^2\}, \quad (2.43)$$

onde \mathbf{w}_i corresponde à i -ésima coluna de \mathbf{W} , $\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij}$, e $g_i(\cdot)$ são funções não-lineares.

Simplificando a expressão (2.43) de maneira matricial, tem-se:

$$J_{NPCA} = E\{\|\mathbf{x} - \mathbf{W}^T \mathbf{g}(\mathbf{W}\mathbf{x})\|^2\}, \quad (2.44)$$

onde $\mathbf{g}(\cdot) = [g_1(\cdot), \dots, g_N(\cdot)]^T$.

Na abordagem de NPCA, as misturas tipicamente passam por uma etapa de branqueamento, a qual pode ser realizada através da PCA. Diante disso, a matriz \mathbf{W} a ser determinada é ortogonal, ou seja, $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. Aplicando essa condição, obtém-se:

$$J_{NPCA}(\mathbf{W}) = \sum_{i=1}^N E\{[y_i - g_i(y_i)]^2\} \quad (2.45)$$

2.6 Separação de Misturas Não-Lineares

Até este momento, temos visto abordagens de separação de fontes no âmbito de misturas lineares. Além disso, apresentamos uma ferramenta, a ICA, capaz de solucionar o problema de separação resultante de maneira satisfatória. Porém, em alguns cenários práticos como análise iônica por arranjos de sensores químicos e comunicações via satélite, há elementos não-lineares no processo de mistura, o que exige uma correspondente adequação da formulação de BSS.

Nesta seção, abordaremos o problema de BSS em misturas não-lineares, com destaque para um caso particular chamado de modelo de mistura com não-linearidade posterior (PNL, do inglês *Post-Nonlinear*).

2.6.1 Misturas Não-Lineares

Em muitas situações práticas, o modelo linear é incapaz de descrever adequadamente o processo de mistura. Portanto, é natural considerar uma extensão não-linear da seguinte forma:

$$\mathbf{x} = \mathbf{f}(\mathbf{s}), \quad (2.46)$$

onde \mathbf{x} é o vetor das misturas, \mathbf{s} é o vetor de fontes e $\mathbf{f}(\cdot)$ é um mapeamento não-linear. Neste trabalho, assumiremos sempre que o modelo de mistura é instantâneo e o número de fontes é igual ao número de misturas.

O mapeamento do sistema separador é representado pela seguinte equação:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \quad (2.47)$$

onde $\mathbf{g}(\cdot)$ corresponde a um mapeamento não-linear. Uma dificuldade importante em modelos não-lineares vem da escolha da estrutura de separação $\mathbf{g}(\cdot)$, pois nem sempre é possível determinar um sistema separador capaz de inverter perfeitamente a ação de $\mathbf{f}(\cdot)$.

2.6.2 ICA Não-Linear

Uma maneira de tratar o problema de separação cega de fontes não-linear é buscarmos a extensão da ideia de ICA para este caso. Porém, quando se aplica a ICA em BSS não-linear, não há garantia da recuperação das fontes. Para entendermos o porquê dessa limitação, vamos expor o conceito de mapeamento trivial.

Um mapeamento $\mathcal{H}(\cdot) = [\mathcal{H}_1(\cdot), \dots, \mathcal{H}_n(\cdot)]$ é considerado trivial se transforma um vetor (original) $\mathbf{r} = [r_1, \dots, r_n]$, com elementos estatisticamente independentes, em um vetor (resultante) cujos elementos também são independentes. Nesse sentido, um mapeamento $\mathcal{H}(\cdot)$ pode ser considerado trivial se, e somente se, a seguinte condição for satisfeita [36]:

$$\mathcal{H}_i(r_1, \dots, r_n) = h_i(r_{\rho(i)}), \quad i = 1, \dots, n \quad (2.48)$$

onde r_1, \dots, r_n são as componentes do vetor original, $h_i(\cdot)$ são funções arbitrárias, ρ corresponde a um operador de permutação sobre o conjunto $\{1, 2, \dots, n\}$ e $h_i(r_{\rho(i)})$ é a i -ésima componente do vetor resultante.

Podemos concluir que um mapeamento é considerado trivial se, e somente se, cada elemento do vetor resultante for uma função exclusiva de um dos elementos r_i , ou seja, as componentes do vetor resultante não podem ser combinações das componentes do vetor original.

Considerando os mapeamentos $\mathbf{f}(\cdot)$ e $\mathbf{g}(\cdot)$ definidos em (2.46) e (2.47), é possível escrever a seguinte relação para as estimativas das fontes:

$$\mathbf{y} = \mathbf{g}(\mathbf{f}(\mathbf{s})) = (\mathbf{g} \circ \mathbf{f})(\mathbf{s}). \quad (2.49)$$

Uma solução para o problema de separação cega de fontes não-linear decorrente da equação (2.49) pode ser obtida considerando que a composição $\mathbf{g} \circ \mathbf{f}$ seja um mapeamento trivial. Logo, nesta situação, a recuperação das fontes também apresentaria a ambiguidade relacionada à permutação de ordem.

Para o caso não-linear genérico pode ocorrer, além dessa permutação a presença de distorção residual não-linear [37]. Mesmo com essa potencial distorção, é possível estimar as fontes se o mapeamento $\mathbf{g} \circ \mathbf{f}$ for trivial. Porém, a estratégia da ICA não-linear não consegue garantir que sempre haverá um mapeamento trivial para representar o sistema separador, pois existem mapeamentos não-triviais que transformam as fontes em vetores independentes, e, ainda assim, as misturam. Deste modo, a resolução deste problema torna-se inviável. Esta dificuldade na solução de BSS não-linear via ICA é consequência de funções não-lineares gerais apresentarem um enorme grau de flexibilidade.

Alguns estudos foram feitos de modo a encontrar modelos para os quais a aplicação da ICA ainda garanta a recuperação das fontes [38],[39] e [37]. Nesse mesmo intuito de encontrar uma solução para o problema de BSS não-linear via ICA, foi proposto por Taleb e Jutten o modelo *Post-Nonlinear* [8]. Desde então este modelo, que será descrito na próxima seção, vem sendo utilizado nos principais estudos da área.

2.7 Mistura com Não-Linearidade Posterior ou *Post-Nonlinear*

Nesta seção, será apresentado ao leitor o modelo de mistura em torno do qual gravita este trabalho. Ainda que o problema de BSS não-linear não admita uma solução geral, o modelo PNL traz perspectivas importantes, pois ainda é possível, neste caso, utilizar a ICA para a separação. O modelo de mistura PNL é constituído por um estágio linear seguido por um conjunto de funções não-lineares que são aplicadas individualmente a cada um dos sinais linearmente misturados. Já o sistema separador é uma versão espelhada do sistema misturador, constituído de um conjunto de não-linearidades $\mathbf{g}(\cdot)$ e de uma matriz \mathbf{W} , como apresentado na Fig. 2.9. Note que a ação das funções não-lineares se dá de modo individual, ou seja, existe uma não-linearidade para cada mistura.

Matematicamente, os sinais observados que formam o vetor \mathbf{x} podem ser representados por:

$$\mathbf{x} = \mathbf{f}(\mathbf{A}\mathbf{s}), \quad (2.50)$$

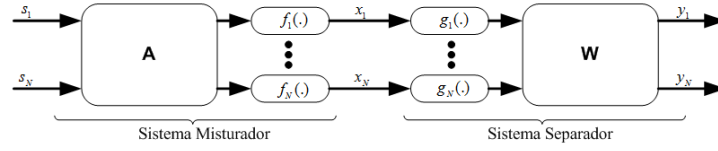


Figura 2.9: Modelo PNL

onde $\mathbf{f}(\cdot) = [f_1(\cdot) \dots f_N(\cdot)]^T$ representa as não-linearidades do sistema misturador e \mathbf{A} representa uma matriz de mistura. Um candidato natural para constituir o sistema separador no caso PNL é descrito por:

$$\mathbf{y} = \mathbf{W}\mathbf{g}(\mathbf{x}), \quad (2.51)$$

onde $\mathbf{g}(\cdot) = [g_1(\cdot) \dots g_N(\cdot)]^T$ representa as funções não-lineares do sistema separador e \mathbf{W} representa uma matriz de separação.

2.7.1 Separabilidade do Modelo PNL

No caso de BSS/ICA linear, para ocorrer a separação, basta adaptar a matriz \mathbf{W} para que os sinais estimados também sejam estatisticamente independentes. No modelo PNL, além de adaptar a matriz \mathbf{W} , é necessário também adaptar cada uma das funções não-lineares presentes no sistema separador. O sistema misturador e separador são descritos conjuntamente pela seguinte expressão:

$$\mathbf{y} = \mathbf{W}\mathbf{g}(\mathbf{f}(\mathbf{A}\mathbf{s})). \quad (2.52)$$

Como no caso linear, a aplicação da ICA ao modelo PNL também requer algumas condições para que ocorra a separação de fontes [8]. De acordo com Taleb e Jutten, as condições de separabilidade no modelo PNL são:

Teorema 2 (Separabilidade do Modelo PNL). *Considere as seguintes hipóteses*

- A matriz \mathbf{A} é inversível, e, de fato, produz uma mistura das fontes, ou seja, há ao menos dois elementos não-nulos em cada coluna e linha dessa matriz;
- As funções $\mathbf{f}(\cdot)$ e $\mathbf{g}(\cdot)$ são monotônicas, e, conseqüentemente, $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ também o é;
- Há, no máximo, uma fonte gaussiana;
- A função densidade de probabilidade conjunta das fontes é diferenciável e sua derivada é contínua em todo seu suporte.

Nessas condições, se os elementos de \mathbf{y} forem estatisticamente independentes, então todos os elementos do vetor $\mathbf{h}(\cdot)$ corresponderão a funções lineares e $\mathbf{WA} = \mathbf{DP}$.

2.8 O Algoritmo de Taleb e Jutten

Taleb e Jutten propuseram um algoritmo de separação PNL baseado na minimização da informação mútua através de uma metodologia via gradiente descendente [8].

Para que determinemos a informação mútua na saída do separador, é preciso obter a seguinte entropia conjunta:

$$H(\mathbf{y}) = H(\mathbf{x}) + E\{\log |J|\}, \quad (2.53)$$

onde J representa o Jacobiano da transformação não-linear entre o vetor de misturas e as saídas do sistema separador. Aplicando (2.53) em (2.11), e levando em conta a estrutura mostrada na Fig. 2.9, obtém-se a seguinte expressão [8]:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{x}) - \log(|\det \mathbf{W}|) - E \left\{ \log \left(\prod_i |g'_i(x_i)| \right) \right\}, \quad (2.54)$$

onde $g'_i(\cdot)$ denota a derivada de $g_i(\cdot)$. Esta relação é válida somente se o sistema separador for inversível.

O gradiente de (2.54) em relação a \mathbf{W} é dado por:

$$\frac{\partial I(\mathbf{y})}{\partial \mathbf{W}} = -E\{\Psi(\mathbf{y})\mathbf{e}^T\} - (\mathbf{W}^T)^{-1}, \quad (2.55)$$

onde $\Psi(\mathbf{y}) = [\psi_{y_1}(y_1), \dots, \psi_{y_N}(y_N)]$ contém as *score functions* de cada y_i , isto é, $\psi_{y_i}(y_i) = (p_{y_i}(y_i))' / p_{y_i}(y_i)$. O vetor $\mathbf{e} = [e_1, \dots, e_N]$ denota as saídas da seção não-linear do sistema separador. Para a estimativa de $\Psi(\mathbf{y})$, Taleb e Jutten utilizaram duas estratégias: expansão de Gram-Charlier e estimação direta da densidade de probabilidade [8].

Considerando funções não-lineares parametrizadas, denotadas por $g_i(\Theta_i, x_i)$, $i = 1, \dots, N$, onde Θ_i corresponde ao um conjunto de parâmetros, o gradiente da informação mútua em relação a Θ_i é dado por:

$$\frac{\partial I(\mathbf{y})}{\partial \Theta_i} = -E \left\{ \frac{\partial \log |g'_i(\Theta_i, x_i)|}{\partial \Theta_i} \right\} - E \left\{ \left(\sum_{k=1}^N \psi_{y_k}(y_k) w_{ki} \right) \frac{\partial g_i(\Theta_i, x_i)}{\partial \Theta_i} \right\}, \quad (2.56)$$

onde w_{ki} corresponde ao elemento (k, i) da matriz \mathbf{W} .

O algoritmo completo de Taleb e Jutten é apresentado na Tab. 2.2. Note que a atualização ocorre

primeiramente na seção não-linear e, em seguida, os novos valores são utilizados na atualização da seção linear.

Tabela 2.2: Algoritmo de Taleb e Jutten

1. Defina os passos referentes às seções não-linear (μ_n) e linear(μ_l);
2. Adaptação da seção não-linear de acordo com a expressão (2.56);

$$\Theta_i \leftarrow \Theta_i - \mu_n \frac{\partial I(\mathbf{y})}{\partial \Theta_i}. \quad (2.57)$$

3. Cálculo da nova saída da seção não-linear: $\mathbf{e} = \mathbf{g}(\Theta, \mathbf{x})$
4. Adaptação da seção linear de acordo com a expressão (2.55).

$$\mathbf{W} \leftarrow \mathbf{W} - \mu_l \frac{\partial I(\mathbf{y})}{\partial \mathbf{W}}. \quad (2.58)$$

5. Estimativa das fontes: $\mathbf{y} = \mathbf{W}\mathbf{e}$.
6. Caso não tenha convergido, voltar ao passo 2.

2.9 Conclusão: Dificuldades em Separação de Misturas PNL

A metodologia que Taleb e Jutten propuseram, vista na Seção 2.7, apresenta dois pontos que requerem especial atenção: o primeiro deles se relaciona a escolha de uma estrutura não-linear genérica, e, ao mesmo tempo, inversível, já o segundo se vincula a multimodalidade da função custo de informação mútua.

Em Duarte *et al.* [1] a questão da multimodalidade foi abordada com o auxílio de uma ferramenta de computação natural, mais especificamente um Sistema Imunológico Artificial. No entanto, não foi a meta do trabalho de Duarte *et al.* realizar uma análise mais sistemática de diferentes metodologias que permitisse verificar com maior precisão a eficiência dessas técnicas à minimização da informação mútua, o que corresponde a uma das metas deste trabalho.

Além disso, não havia sido proposto um arcabouço de projeto das não-linearidades que fosse ao mesmo tempo amplo e factível do ponto de vista de inversibilidade. Neste sentido, vai a segunda e principal contribuição deste trabalho, que é combinar aproximadores monotônicos e algoritmos de otimização bio-inspirados para gerar uma solução ampla do problema PNL no âmbito de minimização da informação mútua.

Tendo apresentado os fundamentos do problema de BSS não-linear, o passo seguinte é analisar as metodologias adotadas para a estimação e otimização da métrica de quantificação de independência escolhida.

Capítulo 3

Otimização da Informação Mútua Utilizando Algoritmos Bio-Inspirados

Problemas de otimização são fundamentalmente caracterizados por situações em que se deseja maximizar ou minimizar uma função de uma ou mais variáveis, havendo ou não restrições. A solução para esses problemas advém dos chamados métodos de otimização, que podem ser divididos em dois grupos: determinísticos e aproximados.

Os métodos de otimização baseados em algoritmos determinísticos, entre os quais se inclui a maioria das ferramentas de otimização não-linear, geram uma sequência determinística de possíveis soluções, usando tipicamente ao menos a primeira derivada da função custo em relação às variáveis a serem otimizadas. Esses métodos apresentam, em geral, garantia de convergência para uma solução localmente ótima, que, entretanto, não é necessariamente um ótimo global. Como nesses métodos a solução encontrada depende fortemente do ponto de partida fornecido, o algoritmo pode convergir para um ótimo local ruim, que pode levar a um mau desempenho quando se lida com funções multimodais, isto é, funções que possuem vários ótimos locais.

Uma alternativa que traz maior chance de encontrar a solução global do problema é recorrer a métodos de otimização denominados aproximados, entre os quais se incluem as *metaheurísticas*. Estas técnicas, quando aplicadas a problemas de otimização, têm como objetivo explorar com uma eficiência maior o espaço de busca, evitando uma parada prematura em ótimos locais e proporcionando, idealmente, a obtenção de melhores soluções. Deste modo, a metaheurística é tipicamente capaz de encontrar soluções viáveis e promissoras dentre todas as possíveis, embora não garanta que a melhor solução será de fato encontrada.

Um ponto fundamental nesse trabalho diz respeito ao problema de otimização da informação mútua associado à construção de um modelo separador PNL, como visto na Seção 2.8. Tal problema se presta à abordagem através da otimização por metaheurísticas, uma vez que a função custo apresenta

características multimodais e é de difícil manipulação.

3.1 Critério de Informação Mútua e Estatísticas de Ordem

Assim como no caso linear, quando se lida com misturas PNL, é possível utilizar um critério de separação baseado na informação mútua. Este critério foi inicialmente proposto por Taleb e Jutten, conforme visto da Seção 2.7.

Repetiremos aqui a expressão da informação mútua para um modelo PNL:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{x}) - \log(|\det(\mathbf{W})|) - E \left\{ \log \left(\prod_i |g'_i(x_i)| \right) \right\}, \quad (2.54)$$

Para que se estabeleça uma função custo de informação mútua, conforme mostra a Equação (2.54), um ponto crucial é a estimação das entropias marginais. Neste trabalho optamos como em [1], por uma abordagem baseada em estatísticas de ordem, a qual será descrita a seguir.

3.1.1 Estimação de Entropia Utilizando Estatísticas de Ordem

A metodologia de estimação de entropia baseada nas chamadas *estatísticas de ordem* apresenta um grau significativo de robustez e uma complexidade computacional tipicamente menor que a associada a métodos baseados na estimação de densidades de probabilidade. Esta metodologia tem sido utilizada com sucesso em separação de misturas lineares e não-lineares [13], [2].

O conceito de estatística de ordem está relacionado ao ordenamento de variáveis aleatórias. Considere o conjunto de variáveis aleatórias Y_1, Y_2, \dots, Y_T . Dado um conjunto de realizações (representadas por y_1, y_2, \dots, y_T), o valor $m = \max y_i$ também representará a realização de uma variável aleatória, que representaremos por $Y_{(T:T)}$. De maneira equivalente, poderíamos considerar o valor mínimo em cada realização, e obtendo assim uma outra variável aleatória $Y_{(1:T)}$. Generalizando esse conceito, chegamos a um conjunto de variáveis aleatórias ordenadas, $Y_{(1:T)} \leq Y_{(2:T)} \leq \dots \leq Y_{(T:T)}$, onde $Y_{(t:T)}$ corresponde à t -ésima estatística de ordem e representa a variável associada ao t -ésimo menor valor das realizações de Y_1, Y_2, \dots, Y_T .

Note que o conceito de estatísticas de ordem também pode ser definido para uma única variável aleatória. Neste caso, cada realização de um conjunto T variáveis aleatórias com distribuições de probabilidade idênticas pode ser vista como T realizações independentes de uma única variável aleatória.

O interesse em se empregar o conceito de estatística de ordem na estimação da entropia está relacionado a sua ligação com a função quantil de uma variável aleatória $Q_Y(\cdot)$, dada por:

$$E\{Y_{(t:T)}\} = Q_Y\left(\frac{t}{T+1}\right), \quad (3.1)$$

onde $t = 1, \dots, T$. A entropia de uma variável aleatória pode ser definida em termos dessa equação, da seguinte maneira:

$$H(Y) = \int_0^1 \log(Q'_Y(u)) du, \quad (3.2)$$

onde Q'_Y denota a derivada da função quantil. Empregando uma aproximação numérica para se calcular a integral em (3.2), obtemos:

$$H(Y) = \sum_{l=2}^L \log\left(\frac{Q_Y(u_l) - Q_Y(u_{l-1})}{u_l - u_{l-1}}\right) \frac{u_l - u_{l-1}}{u_L - u_{l-1}}, \quad (3.3)$$

onde u_1, \dots, u_L é um conjunto de números crescentes no intervalo $[0, 1]$.

A ligação entre a estimação de entropia e as estatísticas de ordem está na relação entre (3.3) e a função quantil. Uma boa estimativa de $Q_Y\left(\frac{t}{T+1}\right)$, chamada de função quantil empírica, pode ser obtida a partir da realização da estatística de ordem t , ou seja, $y_{(t:T)}$.

$$Q_{(y)}\left(\frac{t}{T+1}\right) \approx y_{t:T}, \quad (3.4)$$

onde $t = 1, \dots, T$. Para valores da função quantil que não pertencem ao conjunto $t/(T+1)$, é possível realizar uma interpolação linear, gerando um estimador de entropia simples e eficaz [13].

3.2 Computação Natural e Otimização: Sistemas Imunológicos Artificiais e Inteligência de Enxame

Conforme exposto na Seção 2.9, o problema de otimização da informação mútua é particularmente complexo por dois motivos fundamentais: existência de múltiplos ótimos locais [1], [40] e dificuldade de manipulação da função custo. Por esse motivo, são adotadas neste trabalho meta-heurísticas derivadas da área de computação natural.

Técnicas de otimização baseadas em computação natural têm se mostrado eficientes em termos de capacidade de busca global em problemas de difícil ou impossível tratamento por metodologias clássicas de otimização não-linear [41]. Neste trabalho, optamos por dois algoritmos com inspirações diferentes: um sistema imunológico artificial (AIS, do inglês *Artificial Immune System*), que pode ser

visto como um extensão do algoritmo CLONALG [12] no sentido de lidar com codificação real, e uma implementação de uma técnica de enxame de partículas (*particle swarm*) [42]. O primeiro se baseia em princípios, teorias e modelos dos sistemas imunológicos naturais, enquanto o segundo método se baseia na simulação de comportamento social.

3.2.1 Sistemas Imunológicos Artificiais

Sistemas Imunológicos Artificiais [3] são técnicas inspiradas no funcionamento do sistema de defesa de organismos superiores contra invasores. Essas técnicas podem ter finalidades diversas, como o modelamento de sistemas biológicos e a solução de problemas de engenharia.

No contexto de aplicações práticas de sistemas imunológicos artificiais, um grande destaque está no uso dessas ferramentas em otimização de funções [3] [12]. Nesse caso, uma ideia importante é relacionar o problema de otimização a ser resolvido ao processo evolutivo que ocorre em nosso organismo, que permite que surjam anticorpos adaptados a novos antígenos. Dessa forma, considera-se que cada solução do problema de otimização corresponde à estrutura de um anticorpo e que a função custo a ser otimizada é uma medida de adequação ou *fitness* dessa estrutura a um certo antígeno.

Um algoritmo que pertence ao grupo dos sistemas imunológicos artificiais é o CLONALG [12], que se baseia no *princípio da seleção clonal*. Antes de descrevermos o algoritmo, discutiremos brevemente alguns elementos do funcionamento do sistema imunológico.

De acordo com [3], o objetivo do sistema imunológico é detectar os elementos que são nocivos ao organismo, ou seja, distinguir entre os elementos *próprios* – elementos do organismo – dos *não-próprios* – elementos estranhos do organismo. Porém, esse não é um problema trivial: uma vez que o número de padrões *não-próprios* é bem maior que o número de elementos *próprios*, os recursos disponíveis são escassos e é necessário que, ao mesmo tempo em que está ocorrendo a proteção, o organismo deve continuar funcionando.

Para resolver esse problema, o sistema imunológico desenvolveu uma arquitetura com várias camadas de defesa. A pele funciona como a primeira barreira contra infecções. Uma segunda barreira é a fisiológica, com condições de temperatura e pH impróprias para sobrevivência de certos organismos estranhos. Algumas vezes, no entanto, essas barreiras não são capazes de eliminar os elementos nocivos do nosso organismo, e, quando isso ocorre, eles são combatidos pelo sistema imunológico inato e pelo sistema imunológico adaptativo. O sistema inato consiste basicamente em células capazes de digerir certos tipos de materiais, por exemplo agentes patogênicos ou antígenos desativados. O sistema adaptativo é mais complexo, pois é responsável por uma imunidade que é estabelecida de modo adaptativo durante a vida do organismo. É ao sistema imunológico adaptativo que está relacionado, de certa forma, um processo de otimização.

Os antígenos que entram no corpo devem ser detectados, e as células responsáveis por este re-

conhecimentos são chamadas de linfócitos. Os linfócitos reconhecem os antígenos e unem-se a eles. Esta união ocorre quando uma ligação molecular é estabelecida entre os antígenos e os receptores presentes na superfície do linfócito, conforme apresentado de forma pictórica na Fig. 3.1.

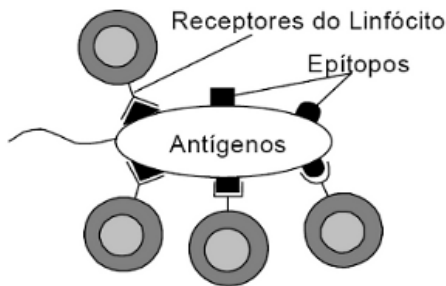


Figura 3.1: União entre receptores e epítopos (baseado em [3])

Note que um linfócito possui receptores em sua superfície, e, assim sendo, pode se unir somente com epítopos estruturalmente relacionados. O número de receptores que se unem aos antígenos irão determinar a *afinidade* ou (*fitness*) do linfócito a um certo antígeno. Os linfócitos somente podem ser ativados por um antígeno se esta afinidade ultrapassar um certo limiar. À medida que este limiar aumenta, o número de tipos de epítopos capazes de ativar o linfócito diminui, isto é, o subconjunto de similaridade torna-se menor. A detecção é aproximada porque é muito difícil evoluir estruturas de receptores que sejam exatamente complementares aos epítopos que o organismo nunca encontrou antes. Um fato importante que devemos salientar é que um único linfócito pode detectar um subconjunto de epítopos, ou seja, um número reduzido de linfócitos é capaz de fornecer proteção contra uma grande variedade de antígenos.

Para que os linfócitos detectem todos os padrões *não-próprios*, é necessário que exista uma grande diversidade de receptores, com limiares de afinidade relativamente baixos. Isto permite que o sistema imunológico detecte praticamente qualquer tipo de antígeno. À medida que os agentes patogênicos entram no organismo, os linfócitos devem detectá-los e eliminá-los rapidamente. Porém, cada antígeno possui diferentes estruturas. Deste modo, o sistema imunológico dispõe de mecanismos que permitem que os linfócitos "aprendam" as estruturas estranhas específicas.

O sistema imunológico evolui e reproduz linfócitos com grande afinidade para antígenos específicos. Este processo é chamado de *maturação de afinidade*, que é um processo darwiniano. Esta maturação ocorre na medula óssea e faz uso de linfócitos chamados de células B. Quando uma célula B é ativada, através da união com os antígenos, ela produz os anticorpos, que são capazes de se unir aos antígenos para deixá-los inativos ou para identificá-los. Após isso, as células B são clonadas, mas as cópias produzidas não são idênticas, pois a clonagem é sujeita a mutações com uma taxa alta. Isto quer dizer que os clones gerados podem ter receptores diferentes da célula mãe. Com isto, as novas

células B podem se ligar aos antígenos. Se elas tiverem uma alta afinidade com os antígenos, elas serão ativadas e clonadas. Quanto maior for a afinidade de uma célula B com os antígenos, maior será a probabilidade de que elas sejam clonadas [41].

Um outro aspecto importante é a chamada *memória imunológica*. Para que as células B não tenham uma vida curta e desapareçam, elas são codificadas. Isto acontece devido ao fato de que, se houver uma nova infecção, a resposta aos agentes patogênicos será mais rápida. A primeira resposta pode custar a acabar com a infecção, e, se o organismo for infectado novamente, terá uma subpopulação adaptada de células B para fornecer uma resposta rápida e específica, chamada de resposta secundária. Esta resposta secundária é tão rápida e eficiente que o organismo não chega a ter ciência de que foi infectado novamente.

3.2.2 Princípio da Seleção Clonal

O princípio da seleção clonal [43] descreve as características básicas de uma resposta imunológica a um antígeno. Em termos simples esse princípio pode ser exposto da seguinte forma:

1. As células filhas são cópias de seus pais (clones) sujeitas a um processo de mutação com altas taxas;
2. Tendem a proliferar as células maduras mais aptas, isto é, aquelas com maior afinidade aos antígenos.

Adaptando tais ideias ao contexto de otimização de uma função custo qualquer, podemos representar cada possível solução do problema como um indivíduo da população de células. No caso de separação de misturas PNL, cada solução é, portanto, composta de um conjunto de parâmetros que definem as não-linearidades e da matriz de separação. A função custo a ser otimizada representa a medida de afinidade entre as células e os antígenos, e, em nosso caso, está relacionada à informação mútua entre as saídas do sistema separador. Além disso, adota-se a ideia de edição de receptores, segundo a qual, após um certo número de iterações, novas soluções são geradas aleatoriamente, o que permite ao algoritmo explorar novas regiões do espaço de busca. O resultado da combinação entre a seleção clonal e inserção de novos indivíduos é um algoritmo com capacidade de busca local e de busca global, ou seja, capaz de, potencialmente, realizar de modo eficiente a otimização de funções multimodais.

3.2.3 Aspectos da Implementação do Algoritmo CLONALG

No algoritmo CLONALG, busca-se a maximização do *fitness*. Porém, estamos interessados na minimização da informação mútua. Além disso, é desejável que o *fitness* assuma sempre valores

positivos. Portanto, em nosso problema, tendo por base a equação (2.54), a função de *fitness* pode ser definida no seguinte modo:

$$J_{Fit} = \frac{1}{1 + \sum_i H(y_i) - \log(|\det \mathbf{W}|) - E\{\log(\prod_i |g_i^i(x_i)|)\}}. \quad (3.5)$$

As entropias marginais são calculadas de acordo com a abordagem de estatísticas de ordem, descrita na Seção 3.1.1.

Na Tab. 3.1, apresentamos um pseudo-código que ilustra o funcionamento do CLONALG.

Tabela 3.1: Descrição do Algoritmo CLONALG

<ol style="list-style-type: none"> 1. Inicialize aleatoriamente a população , que é composta de(N) indivíduos, 2. Determine o <i>fitness</i> (custo) J_{Fit} de cada indivíduo; 3. Enquanto o número máximo de gerações não é atingido, faça: <ol style="list-style-type: none"> (a) Crie N_c clones para cada indivíduo; (b) Mantenha o indivíduo original a aplique um processo de mutação para cada clone seguindo a equação: $c' = c + \alpha Y(0, 1), \quad (3.6)$ $\alpha = \frac{1}{\beta} \exp(-J_{Fit}), \quad (3.7)$ <p>onde c e c' representam o clone modificado pela mutação e o original, respectivamente. $Y(0, 1)$ corresponde a uma variável aleatória Gaussiana com média zero e variância unitária e β é um parâmetro de controle;</p> (c) Avalie o <i>fitness</i> de cada novo indivíduo e mantenha na população somente a melhor solução de cada grupo formado pelo indivíduo e seus clones mutados; (d) A cada t iterações, elimine os m elementos da população com os menores valores de <i>fitness</i> e os substitua por indivíduos gerados aleatoriamente; 4. Retorne ao passo 3.

O processo de mutação está representado no passo **3b** da Tab. 3.1, em que as cópias do vetor de parâmetros estão sujeitas a uma modificação proporcional ao valor de *fitness*. Nessa etapa, juntamente

com o passo **3c**, o algoritmo estabelece um eficiente mecanismo de busca local, apesar de haver também um potencial de busca global. Finalmente, na etapa **3d**, há um aumento determinante no potencial de busca global, causado pela inserção de indivíduos aleatórios na população, o que aumenta a diversidade populacional e permite uma maior exploração no espaço de busca.

3.3 Otimização Baseada em Enxames de Partículas

O estudo do comportamento de animais que vivem em sociedade é uma riquíssima fonte de ideias e estratégias para a solução de diversos tipos de problemas de interesse prático [41]. O algoritmo enxame de partículas (PS, do inglês *Particle Swarm*), apresentado nesta seção, tem como motivação a habilidade social inerente a certas espécies. Essa técnica de otimização foi desenvolvida na década de 90, mais precisamente em 1995, por James Kennedy e Russel Eberhart [44]. Uma inspiração fundamental para a abordagem é a capacidade de uma população de indivíduos interagir com o ambiente e com outros indivíduos da mesma espécie, em particular com seus vizinhos. Esta interação, discutida em [42], se associa a dois tipos de informações que um indivíduo leva consigo. O primeiro deles consiste no aprendizado individual e o segundo diz respeito ao conhecimento compartilhado com os outros indivíduos da espécie. Portanto, a decisão que um indivíduo deve tomar é função de como o sucesso desta decisão o afetou no passado, e, além disso, é afetada também pelas influências sociais.

Em [42], são utilizados três princípios para resumir o processo cultural de adaptação do algoritmo de enxame de partículas: *Avaliar*, *Comparar* e *Imitar*, que detalhamos a seguir:

- **Avaliar** - Este princípio diz respeito à capacidade dos indivíduos de analisar o ambiente. Isto permite que esses indivíduos possam quantificar o grau de convivência em relação aos outros.
- **Comparar** - Após a primeira análise, o indivíduo tende a fazer comparações com outros da sua espécie, permitindo que isto sirva de motivação para aprender ou mudar.
- **Imitar** - Quando um indivíduo é melhor, o fato de o imitar possibilita uma melhora diante da sociedade.

3.3.1 Aspectos da Implementação do Algoritmo Enxame de Partículas

No algoritmo de enxame de partículas, indivíduos buscam soluções para um dado problema de acordo com sua própria experiência e com a experiência dos outros. Como foi descrito anteriormente, os indivíduos avaliam a si mesmos, comparam-se com seus vizinhos e imitam somente aqueles que são superiores a eles [41].

Na implementação clássica de métodos de otimização baseados em enxames de partículas, cada solução do problema a resolver corresponde a uma partícula, que é vista como um ponto no espaço de busca. Cada partícula ou indivíduo i , em uma dada iteração t do algoritmo, está associada a um vetor posição $\mathbf{x}_i(t)$ e a um vetor velocidade $\mathbf{v}_i(t)$. A função custo (3.5) é utilizada para a avaliação (*fitness*) de cada partícula.

As posições e velocidades das partículas são determinadas por suas próprias experiências, tanto quanto pela observação de seus vizinhos, o que significa que a melhor posição pessoal e a melhor posição dos vizinhos definem cada expressão de atualização.

Como pode ser observado na Tab. 3.2, o algoritmo possui uma estrutura simples. Outro fator que chama a atenção é o pequeno número de parâmetros que devem ser ajustados *a priori*.

Tabela 3.2: Descrição do Algoritmo de Enxame de Partículas

1. Inicializar as partículas, escolher os valores de velocidade máxima e mínima (V_{max} e V_{min}) e calcular o *fitness* (custo) inicial de todas as partículas;
2. Para $i = 1$ até N , fazer:
 - (a) se $f(\mathbf{x}_i) > f(\mathbf{p}_i)$ então $\mathbf{p}_i = \mathbf{x}_i$;
 - (b) $g = i$;
 - (c) se $f(\mathbf{p}_j) < f(\mathbf{p}_g)$, para $j = \text{índices dos vizinhos}$, então $g = j$;
 - (d) Atualizar velocidade e posição:

$$\Delta \mathbf{x}_i \leftarrow \Delta \mathbf{x}_i + \varphi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i) + \varphi_2 \otimes (\mathbf{p}_g - \mathbf{x}_i) \quad (3.8)$$

$$\Delta \mathbf{x}_i \in [V_{min}, V_{max}] \quad (3.9)$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta \mathbf{x}_i \quad (3.10)$$

3. Repetir o procedimento a partir do passo 2 até que o critério de parada seja satisfeito.

Na Equação (3.8), em que ocorre a atualização da velocidade, temos que φ_1 e φ_2 representam vetores aleatórios com valores positivos derivados de uma distribuição uniforme entre 0 e AC_1 e entre 0 e AC_2 , respectivamente, sendo AC_1 e AC_2 chamados de constantes de aceleração. O segundo termo do segundo membro da Equação (3.8) é proporcional à diferença entre a melhor posição (\mathbf{p}_i)

já visitada por um indivíduo e a sua posição atual (\mathbf{x}_i) e o último termo é proporcional à diferença entre a melhor posição ocupada por um vizinho (\mathbf{p}_g) e a posição atual. O símbolo \otimes representa a multiplicação de vetores elemento a elemento. Para limitar a mudança de posição de um indivíduo, definem-se dois valores, V_{min} e V_{max} , que não podem ser excedidos por nenhum elemento de $\Delta\mathbf{x}$, conforme (3.9).

3.4 Conclusões

Tendo visto os aspectos relacionados à otimização pertinentes ao arcabouço geral para solucionar o problema de separação de misturas PNL que norteia este trabalho, estamos prontos a investigar os pontos relacionados à estrutura do sistema separador. Como já visto na Seção 2.9, a dificuldade neste caso é que o uso de estruturas sem restrições pode violar as condições de separabilidade do modelo PNL. No capítulo seguinte, proporemos uma abordagem que evita tal dificuldade sem perda do caráter genérico desejado.

Capítulo 4

Estruturas Monotônicas

Quando se lida com métodos de separação para modelos PNL, surge uma dificuldade que se encontra na parametrização das funções $\mathbf{g}(\cdot)$ de modo a "anular o efeito" das não-linearidades $\mathbf{f}(\cdot)$. Este problema se deve ao fato de que funções não-lineares sem restrição apresentam um enorme grau de flexibilidade e nem sempre permitem que se determine com eficiência um mapeamento capaz de inverter a ação do sistema misturador. Isso pode ser consequência, por exemplo, de uma violação das condições de separabilidade do modelo PNL, vistas na Seção 2.7.1.

Uma solução natural para esse problema vai no sentido de diminuir essa flexibilidade, limitando os modelos do sistema separador a uma classe de funções não-lineares que obedecem às condições de separabilidade do modelo PNL.

No âmbito dessa classe estão as estruturas monotônicas com capacidade de aproximação universal. A opção por estas estruturas possui o objetivo de levar a solução do modelo PNL a um contexto de maior generalidade, visto que elas são capazes, em tese, de inverter qualquer não-linearidade monotônica. As bases de uma proposta nesse sentido serão apresentadas nesse capítulo, em que foram utilizadas três estruturas: polinômios, a rede neural de Sill e uma rede neural de múltiplas camadas, todas com restrição de monotonicidade.

4.1 Polinômios

Uma solução natural e simples quando se lida com o problema de aproximação de funções é empregar polinômios. Um aspecto relevante dessas estruturas é que elas combinam um expressivo potencial de aproximação à simplicidade advinda do fato de serem lineares nos parâmetros.

Em um dos trabalhos de Jutten [9], bem como no de Duarte *et al.* [1], utilizaram-se polinômios com restrição de monotonicidade, mas como soluções específicas, ou seja, *ad hoc*. Neste trabalho, o enfoque será distinto: pretende-se generalizar o uso dos polinômios no sentido de tratá-los como

aproximadores monotônicos gerais.

Podemos definir um polinômio de grau N sem restrição da seguinte forma:

$$y = \sum_{n=0}^N a_n x^n, \quad (4.1)$$

onde o conjunto de valores a_n representa os coeficientes do polinômio. Evidentemente, o polinômio da Equação (4.1) não apresenta ainda a restrição de monotonicidade, a qual pode ser incluída se for adotado um modelo simplificado da seguinte maneira:

$$y = \sum_{n=1}^N a_n x^{2n-1}. \quad (4.2)$$

com a restrição de que $a_n \geq 0$. Observando a equação (4.2), vemos que a derivada de y com respeito a x será composta apenas de monômios de grau par com coeficientes não negativos, o que garante a monotonicidade. Perceba também que, por simplicidade, não consideramos um termo de *offset*.

4.2 Rede Monotônica de Sill

Outra estrutura considerada neste trabalho é a rede neural monotônica proposta por Joseph Sill em 1998 [10], que possui capacidade de aproximação universal [45] de mapeamentos monotônicos. Nessa rede, na qual a monotonicidade é garantida por certas restrições impostas aos pesos sinápticos, geram-se superfícies lineares por partes através do cálculo de máximos e mínimos sobre grupos de hiperplanos. Sua arquitetura possui três camadas, com uma camada oculta, como mostra a Fig. 4.1.

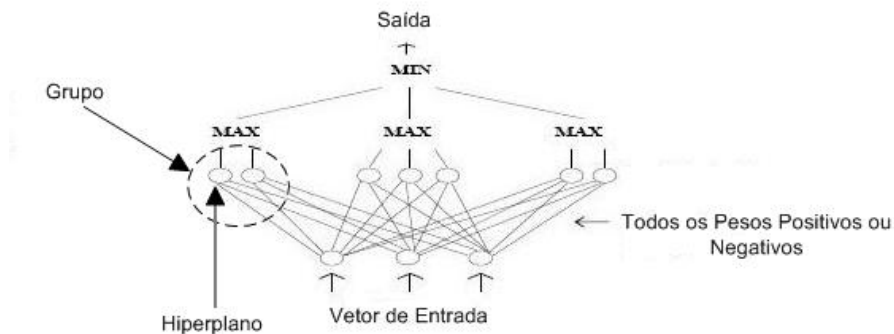


Figura 4.1: Arquitetura da Rede Monotônica de Sill

De acordo com a Fig. 4.1, a camada intermediária de unidades desta rede monotônica calcula diferentes combinações lineares do vetor de entrada. Ainda de acordo com a estrutura desta rede

monotônica, as unidades dessa camada são divididas em alguns grupos, sendo que o número de hiperplanos em cada grupo não é necessariamente o mesmo. A partir deste particionamento surge a camada de saída da rede, que tem como objetivo calcular o mínimo dentre todos os grupos.

Um dado grupo k é constituído de h_k hiperplanos, sendo estes definidos por $\mathbf{w}^{(k,1)}, \mathbf{w}^{(k,2)}, \dots, \mathbf{w}^{(k,h_k)}$ e pelos offsets $t^{(k,1)}, t^{(k,2)}, \dots, t^{(k,h_k)}$. Deste modo, a saída do grupo, g_k , é dada por:

$$g_k(\mathbf{x}) = \max_j \mathbf{w}^{(k,j)} \mathbf{x} - t^{(k,j)}, \quad (4.3)$$

onde $1 \leq j \leq h_k$. Assim sendo, a saída da rede y equivale ao mínimo valor entre todos os grupos, ou seja:

$$y = \min_k g_k(\mathbf{x}). \quad (4.4)$$

No trabalho de Sill [10], define-se o termo *ativo* para denotar grupos e hiperplanos que participam da construção de uma determinada saída. Por exemplo, chamaremos um grupo l de ativo em \mathbf{x} se:

$$g_l(\mathbf{x}) = \min_k g_k(\mathbf{x}), \quad (4.5)$$

isto é, se esse grupo determinou a saída da rede para aquela entrada. De maneira similar, diremos que o hiperplano é ativo em \mathbf{x} se seu grupo também é ativo em \mathbf{x} e este hiperplano é o máximo sobre todos os hiperplanos no grupo.

4.3 Perceptron de Múltiplas Camadas com Restrição de Monotonicidade

Outra estrutura escolhida para compor as não-linearidades no sistema separador de um modelo PNL foi uma rede Perceptron de Múltiplas Camadas (MLP, do inglês *Multi-Layer Perceptron*). A estrutura de uma rede MLP consiste de um conjunto de unidades sensoriais que formam a camada de entrada, uma ou mais camadas intermediárias (ou escondidas) e uma camada de saída, como pode ser visto na Fig. 4.2. É importante ressaltar que uma arquitetura desse tipo pode ser vista como uma ferramenta prática para realizar mapeamentos dotada de capacidade de aproximação universal [45]. Uma particularidade deste trabalho é que, como as funções não-lineares devem possuir restrições de monotonicidade para que o resultado final seja obtido de maneira eficiente, foi preciso adotar a rede MLP monotônica desenvolvida por Bernhard Lang [11], denominada MONMLP - *Monotonic Multi-Layer Perceptron* ou Perceptron de Múltiplas Camadas Monotônico.

Matematicamente, uma MONMLP com I entradas, uma primeira camada oculta com H nós, uma

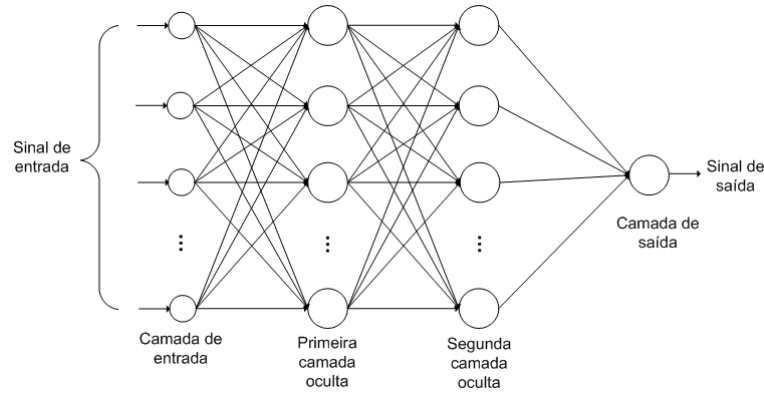


Figura 4.2: Arquitetura da uma Rede Neural MLP.

segunda camada oculta com L nós e uma saída pode ser representada, à primeira vista, como uma MLP padrão:

$$\hat{y}(\mathbf{x}) = w_b + \underbrace{\sum_{l=1}^L w_l \tanh \left(\underbrace{w_{b,l} + \sum_{h=1}^H w_{lh} \tanh \left(\underbrace{w_{b,h} + \sum_{i=1}^I w_{hi} x_i}_{\theta_2} \right)}_{\theta_1} \right)}_{\theta_1} \quad (4.6)$$

Esta rede MLP assegura um comportamento de monotonicidade crescente com respeito à entrada $x_j \in \mathbf{x}$ se:

$$\frac{\partial \hat{y}}{\partial x_j} = \sum_{l=1}^L w_l \cdot \underbrace{(1 - \theta_1^2)}_{>0} \cdot \sum_{h=1}^H w_{lh} \cdot \underbrace{(1 - \theta_2^2)}_{>0} \cdot w_{hj} \geq 0. \quad (4.7)$$

Analisando a Equação (4.7) percebe-se que a derivada da tangente hiperbólica é sempre positiva, e, logo, a condição suficiente para a monotonicidade crescente para uma entrada x_j e a saída \hat{y} dada por:

$$w_l, w_{lh}, w_{hj} \geq 0, \forall l, h. \quad (4.8)$$

Para garantir a monotonicidade decrescente, para esta mesma entrada, a restrição que deve ser feita é a seguinte:

$$w_l, w_{lh}, w_{hj} \leq 0, \forall l, h. \quad (4.9)$$

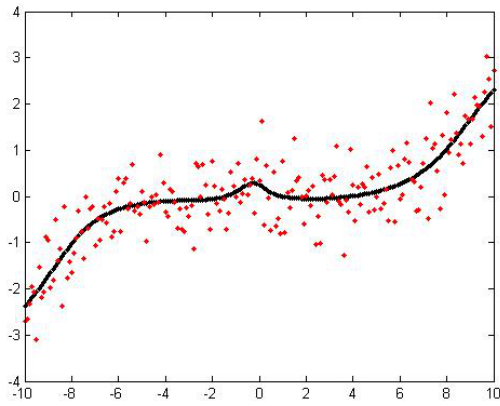
Analisando a Equação (4.7), percebemos que, se todos os pesos relacionados à entrada x_j forem maiores ou iguais a zero, a restrição de monotonicidade (crescente) pode se tornar mais simples:

$$w_{hj} \geq 0, \forall h. \quad (4.10)$$

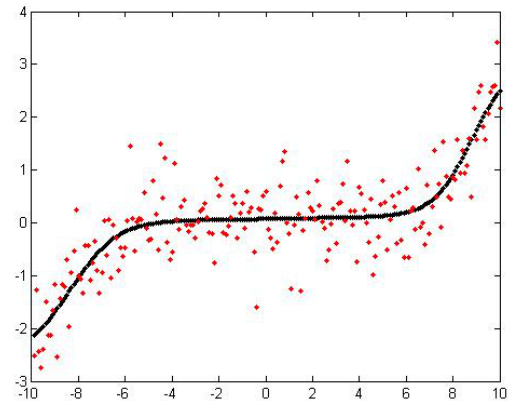
Se o sinal for invertido em (4.10), garante-se monotonicidade decrescente.

Para fazer um contraste entre as capacidades de aproximação dos dois tipos de MLP, é dado a seguir um exemplo simples de regressão não-linear, no qual queremos encontrar a melhor curva que se ajuste, na presença de ruído, aos pontos gerados pela equação:

$$y = ax + bx^3 + cx^5. \quad (4.11)$$



(a) MLP sem Restrição



(b) MLP com Restrição

Figura 4.3: MLP *sem* Restrição de Monotonicidade X MLP *com* Restrição de Monotonicidade

A Fig. 4.3 apresenta os mapeamentos obtidos pela rede MLP sem restrição e com restrição. Observando a Fig. 4.3(a), percebemos que há um "pico" na curva ajustada, o que é consequência da excessiva flexibilidade de monotonicidade da rede MLP, que fez com que ela, devido ao ruído, tivesse um *sobreajuste* ou *sobretreinamento*. Quando isto ocorre, a rede MLP perde a habilidade de generalizar, o que ocasiona um mau desempenho para padrões "não vistos". Por outro lado, com a restrição de monotonicidade, os dados de treinamento foram ajustados adequadamente, resultando em uma boa generalização para este caso específico, como mostrado na Fig. 4.3(b).

4.4 Conclusões

Feita a exposição das estruturas monotônicas aptas a fazerem o papel de não-linearidades do sistema separador PNL, temos em mãos todos os elementos que constituem a proposta que motiva

este trabalho. Deste modo, passaremos à etapa de avaliação desses elementos em diversos cenários representativos.

Capítulo 5

Resultados

Nos capítulos anteriores, apresentamos os fundamentos do problema de BSS, bem como os elementos de uma metodologia para separação de misturas PNL baseada em otimização bio-inspirada da informação mútua e no uso de estruturas monotônicas. Nesse capítulo, essa concepção será testada em diferentes cenários, de tal modo que seja possível averiguar a sua efetividade. Primeiramente, concentraremos nossa atenção no problema de otimização, para, em seguida, passar a uma análise mais geral. A Fig. 5.1 apresenta um esquema de como os resultados serão apresentados.

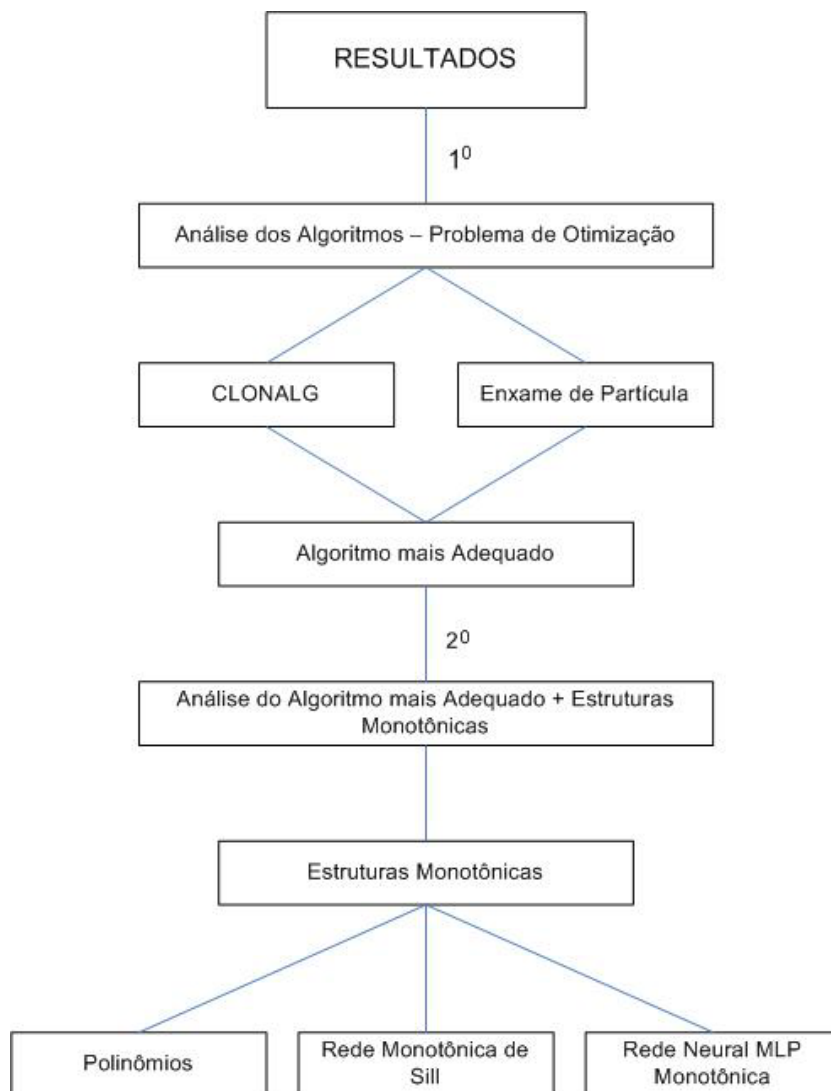


Figura 5.1: Procedimento da Análise de Resultados

5.1 Análise do Emprego de Ferramentas Bio-Inspiradas no Âmbito da Metodologia de Duarte *et al.* [1]

Conforme exposto, num primeiro momento, dedicar-nos-emos a uma análise comparativa de ferramentas de otimização bio-inspiradas num cenário estrutural análogo ao estudado em Duarte *et al.* [1]. Nesse trabalho, foi usada a rede imunológica opt-aiNet [3] juntamente com não-linearidades polinomiais, e, portanto, pareceu-nos oportuno realizar testes mais extensivos com outras abordagens, um algoritmo imunológico menos complexo, o CLONALG [12], e um algoritmo de enxame de partículas [44]. A comparação se dará em dois cenários, com duas e três fontes, exatamente como em [1].

Primeiro Cenário

No primeiro cenário, consideramos a separação de duas fontes uniformemente distribuídas no intervalo $[-1,1]$ e misturadas pelo seguinte sistema PNL:

$$A = \begin{bmatrix} 1 & 0,6 \\ 0,5 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= \tanh(e_1) \\ f_2(e_2) &= \sqrt[5]{e_2} \end{aligned} \quad (5.1)$$

O sistema separador é constituído de uma matriz quadrada \mathbf{W} e de funções não-lineares polinomiais de quinto grau do tipo $y = ax^5 + bx^3 + cx$, com a restrição de monotonicidade exposta na Seção 4.1.

Para realizar os experimentos em ambos os cenários, os parâmetros dos algoritmos bio-inspirados, foram ajustados por meio de uma série de experimentos preliminares, tendo em vista a obtenção de um bom compromisso entre parcimônia e precisão. Para o primeiro cenário, os parâmetros do algoritmo CLONALG foram ajustados conforme a Tab. 5.1.

Tabela 5.1: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	10
Número de Clones (N_c)	7
Regulador de Amplitude de Mutação (β)	60
Número de Indivíduos Reinseridos	2
Número Máximo de Gerações	10000

Os parâmetros do algoritmo de enxame de partículas foram ajustados conforme a Tab. 5.2.

Tabela 5.2: Parâmetros do Algoritmo Enxame de Partícula

Parâmetro	Valor
Número de Indivíduos (N)	60
Constante de Aceleração (AC_1 e AC_2)	2.05
Limites de Velocidade de um Indivíduo	$V_{min} = -0.1$ e $V_{max} = 0.1$
Número Máximo de Gerações	4490

Neste cenário, foram realizados dois experimentos diferentes. No primeiro deles, executamos os algoritmos 25 vezes considerando exatamente as mesmas realizações das fontes. Na segunda bateria de experimentos, efetuamos 50 testes com diferentes realizações das fontes. Conduzimos estes dois

tipos de testes com o objetivo de verificar se, na média, os algoritmos eram capazes de atuar com eficiência no processo de otimização da informação mútua.

Na Tab. 5.3, apresentamos os valores de erro quadrático médio (EQM) para os dois experimentos. Vale salientar que os valores de EQM foram calculados após a normalização da potência de todos os sinais.

Tabela 5.3: EQM - Primeiro Cenário

Quantidade de Simulações	EQM($\times 10^{-2}$)	y_1	y_2
25	CLONALG	0,11	0,65
	Swarm	0,73	0,52
50	CLONALG	0,77	0,43
	Swarm	0,62	0,47

Verificamos na Tab. 5.3, primeiramente, que ambas as técnicas são capazes de atingir um nível médio de erro bastante reduzido, o que indica a sua eficiência em termos de separação de misturas PNL. Nota-se ainda que o desempenho de ambas as técnicas foi aproximadamente equivalente.

Na Fig. 5.2(a), apresentamos a distribuição conjunta das misturas.¹ Para esta situação, considerou-se, na etapa de treinamento, um conjunto de 2000 amostras dos sinais misturados. Após o ajuste do sistema separador através do CLONALG, foi possível obter boas estimativas das fontes, fato que pode ser averiguado na Fig. 5.2(b), que mostra que a distribuição conjunta das estimativas obtidas ficou próxima à de uma uniforme e para sinais independentes, indicando que o sistema separador, com essa estrutura, foi capaz de recuperar os sinais. Não obstante, ainda nesta figura, é possível observar uma certa distorção residual não-linear, decorrente da limitação estrutural inerente ao problema, pois é impossível inverter perfeitamente a ação da tangente hiperbólica através de um polinômio.

Esta análise, apresentada no cenário de duas fontes, revela a eficiência dos métodos de busca e indica que as duas técnicas analisadas são capazes de obter um desempenho não inferior ao da abordagem de Duarte *et al.* [1].

Segundo Cenário

No segundo cenário, consideramos a tarefa de separar três fontes uniformemente distribuídas no intervalo $[-1,1]$. Neste caso, adotamos o seguinte modelo de mistura PNL:

¹Não serão apresentados os gráficos para o algoritmo de enxame de partículas, pois, neste cenário de duas fontes, este algoritmo teve desempenho semelhante ao CLONALG também no que diz respeito às figuras.

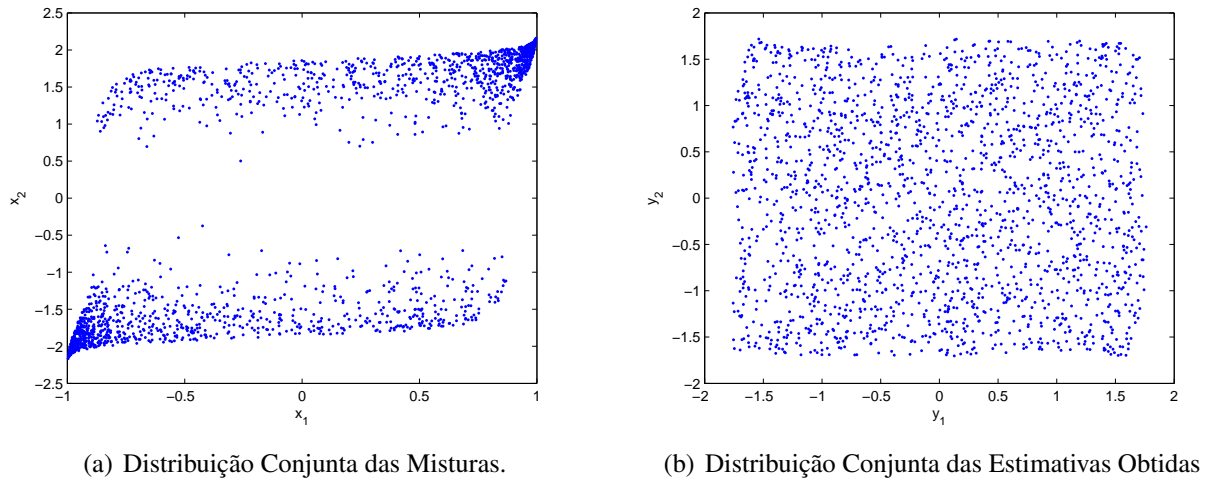


Figura 5.2: Resultados - Primeiro Cenário (CLONALG)

$$A = \begin{bmatrix} 1 & 0,6 & 0,5 \\ 0,5 & 1 & 0,4 \\ 0,4 & 0,6 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= 2\sqrt[3]{e_1} \\ f_2(e_2) &= 2\sqrt[3]{e_2} \\ f_3(e_3) &= 2\sqrt[3]{e_3} \end{aligned} \quad (5.2)$$

Os parâmetros escolhidos para os algoritmos CLONALG e enxame de partículas se encontram nas Tabs. 5.4 e 5.5, respectivamente. As não-linearidades polinomiais de separação foram mantidas, ou seja, também são polinômios de quinto grau com restrição de monotonicidade.

Tabela 5.4: Parâmetros do Algoritmo CLONALG no Segundo Cenário

Parâmetro	Valor
Número de Indivíduos (N)	50
Número de Clones (N_c)	15
Regulador de Amplitude de Mutação (β)	50
Número de Indivíduos Reinseridos	8
Número Máximo de Gerações	10000

Na Tab. 5.6, são apresentados os valores de EQM associados à estimação de cada fonte para os dois testes experimentais.

Tabela 5.5: Parâmetros do Algoritmo Enxame de Partícula no Segundo Cenário

Parâmetro	Valor
Número de Indivíduos (N)	200
Constante de Aceleração (AC_1 e AC_2)	2.05
Limites de Velocidade de um Indivíduo	$V_{min} = -1$ e $V_{max} = 1$
Número Máximo de Gerações	13400

Tabela 5.6: EQM - Segundo Cenário

Quantidade de Simulações	EQM($\times 10^{-2}$)	y_1	y_2	y_3
25	CLONALG	0,42	0,54	0,77
	Swarm	2,12	0,95	0,66
50	CLONALG	0,97	4,10	4,69
	Swarm	23,71	18,62	32,18
Duarte <i>et al.</i> [1]	opt-aiNet	3,35	1,10	1,50

Analizando os valores da Tab. 5.6, podemos considerar que o CLONALG teve desempenho - melhor, tendo atingido níveis de EQM aproximadamente uma ordem de grandeza menores que os relatados em Duarte *et al.* [1]. Apesar disso, o algoritmo enxame de partículas teve um desempenho bastante razoável, com erros médios, mesmo no pior caso, comparáveis aos obtidos em [1]. Vale ainda ressaltar que, em particular, no caso com 50 simulações, os valores mais elevados de EQM médios obtidos com o enxame de partículas são devidos à não convergência em apenas 8 casos. De qualquer forma, podemos considerar a técnica imunológica como a mais robusta no cenário em questão.

De um modo geral, podemos afirmar que ambas as metodologias bio-inspiradas se mostraram adequadas ao problema de separação formulado com base em estatísticas de ordem e numa medida de informação mútua. Os resultados, obtidos em dois cenários diferentes, revelaram que os métodos são ferramentas de busca robustas e eficientes no que se refere ao problema em questão, mas com certa superioridade ao algoritmo CLONALG, principalmente no cenário de separar três fontes.

5.2 Análise do Uso de Estruturas Monotônicas no Modelo PNL

Nesta seção, serão apresentados os resultados da segunda e principal contribuição deste trabalho, que compreende a aplicação de estruturas monotônicas no ajuste das funções não-lineares $\mathbf{g}(\cdot)$ do sistema separador no modelo PNL juntamente com a abordagem de otimização bio-inspirada da informação mútua. Como visto na Seção 2.9, a motivação de estabelecer um arcabouço desse tipo

provém da busca de uma solução geral, que respeite as condições de separabilidade do modelo PNL, e, ao mesmo tempo, seja capaz de lidar com um ampla gama de modelos de mistura. Empregaremos como exposto no Cap. 4 três estruturas monotônicas com capacidade de aproximação universal: polinômios, rede monotônica de Sill e uma rede neural MLP monotônica. Discutiremos os resultados obtidos com cada estrutura em separado.

5.2.1 Resultados – Polinômios

Para avaliar o desempenho da proposta de separação no caso de uma estrutura monotônica com polinômios, realizamos simulações em dois cenários distintos, com duas e três fontes. O desempenho das soluções obtidas foi medido de acordo com a relação sinal-interferência (SIR, do inglês *Signal-to-Interference Ratio*), que matematicamente é dada por:

$$\text{SIR}_i = 10 \log_{10} \left(\frac{E\{s_i^2\}}{E\{(s_i - \hat{s}_i)^2\}} \right), \quad (5.3)$$

onde s_i denota a fonte original e \hat{s}_i é a sua estimativa após a normalização de média e variância. Em termos de SIR, podemos dizer que, a partir de aproximadamente 12dB, começam a ser alcançados resultados satisfatórios para o modelo PNL.

Primeiro Cenário

No primeiro cenário, consideramos a separação de duas fontes uniformemente distribuídas no intervalo $[-1,1]$, sendo válido o seguinte modelo de mistura.

$$A = \begin{bmatrix} 1 & 0,6 \\ 0,5 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= \tanh(e_1) \\ f_2(e_2) &= \tanh(e_2) \end{aligned} \quad (5.4)$$

De modo análogo ao descrito na Seção 5.1, houve alguns experimentos preliminares para que os parâmetros do algoritmo CLONALG fossem ajustados de maneira a obter resultados satisfatórios. Deste modo, após alguns testes, os parâmetros do algoritmo CLONALG foram definidos conforme mostra a Tab. 5.7.

Observando a Tab. 5.7 vemos, na quarta linha, que há dois valores para Regulador de Amplitude de Mutação (β): isto significa que, nas primeiras 5000 gerações, o β é igual a 60, e, nas últimas 5000 gerações, o β é igual a 120. Além disso, como descrito na Seção 3.2.3, o CLONALG possui a capacidade de reinserir novos indivíduos no processo de otimização, o que justifica a penúltima linha da tabela. Finalizando, vale ressaltar que a quantidade de gerações executadas pelo algoritmo foi de 10000 para este caso. Para garantir que a inversa das funções não-lineares fosse bem aproximada,

Tabela 5.7: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	30
Número de Clones (N_c)	8
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	5
Número Máximo de Gerações	10000

optamos, com base em testes preliminares, por um polinômio monotônico de 4 coeficientes, ou seja, de grau sete, o que fez com que houvesse um total de 12 parâmetros a serem ajustados. A Fig. 5.3 apresenta a distribuição conjunta dos dados observados (misturados) e a distribuição após a separação, contendo 1000 amostras dos sinais observados.

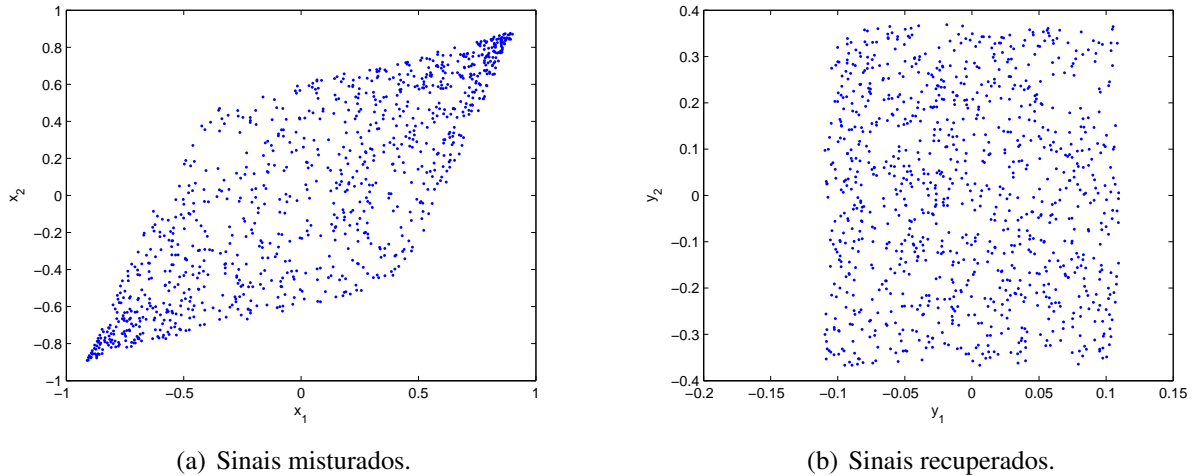


Figura 5.3: Distribuição dos sinais (a) observados e (b) recuperados.

Visualmente, ao analisar as Figuras 5.3 e 5.4, verificamos que, para este caso, o modelo PNL conseguiu recuperar as fontes, visto que, na Fig. 5.3(b), as estimativas das fontes estão próximas de uma distribuição conjunta uniforme para sinais independentes. Note, porém, que existe uma pequena distorção residual, pois os polinômios não conseguem inverter totalmente as tangentes hiperbólicas do sistema misturador, além da ambiguidade de escala que é inevitável quando se lida com ICA. Outro ponto importante nesta análise está ilustrado na Fig. 5.4², no qual comprovamos que os polinômios dão origem a uma condição próxima à ideal, em que a composição $\mathbf{f}_i(\cdot)$ e o polinômio $\mathbf{g}_i(\cdot)$ é um

²Vale ressaltar que as figuras relacionadas à composição das função f_i e g_i , em todos os testes, têm cunho meramente analítico, não sendo passíveis de avaliação em situações práticas

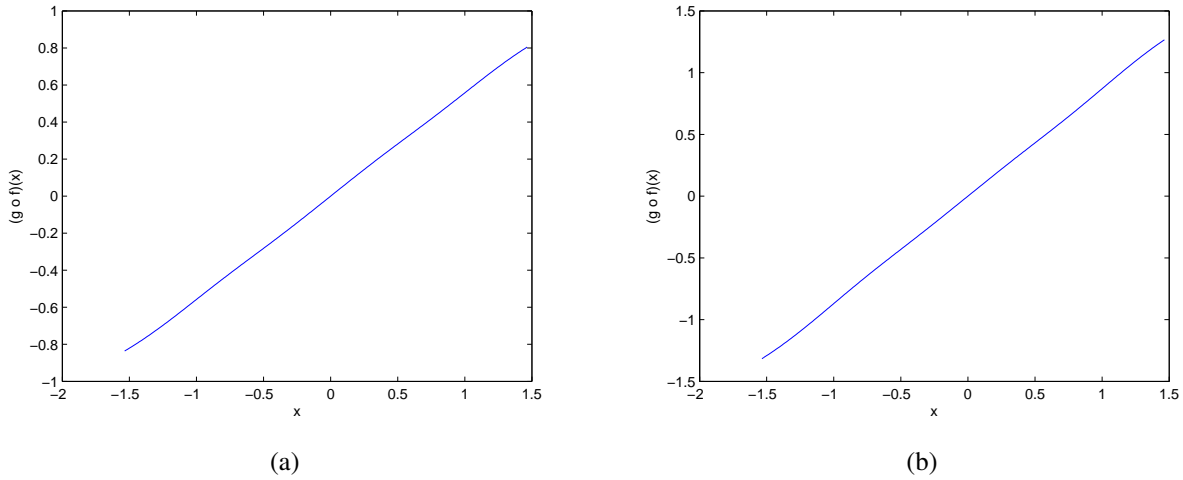


Figura 5.4: Composição das funções f_i e g_i para as duas fontes.

mapeamento muito próximo do linear. A eficiência dos polinômios ainda é comprovada através da SIR, que foi de 35,5 dB e 34,2 dB para as fontes 1 e 2, respectivamente.

Segundo Cenário

Para avaliar o desempenho da estrutura monotônica com polinômios em um cenário mais complexo, optamos por um cenário com três fontes. Nesse caso, devemos ter em mente que o número de parâmetros relacionados à seção não-linear da estrutura de separação cresce linearmente com o número de fontes, mas o número de parâmetros da matriz de separação \mathbf{W} cresce de maneira quadrática, o que representa um aumento considerável na dimensão do espaço de busca dos parâmetros e, conseqüentemente, no custo computacional para que se obtenham boas soluções. Por esse motivo, os parâmetros do algoritmo mostrados na Tab. 5.8, apontam para a necessidade de amostrar mais vezes a função custo.

Tabela 5.8: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	60
Número de Clones (N_c)	15
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	12
Número Máximo de Gerações	10000

Como no primeiro cenário, as fontes estão uniformemente distribuídas no intervalo $[-1,1]$, mas o sistema misturador é mais complexo, como mostrado a seguir:

$$A = \begin{bmatrix} 1 & 0,6 & 0,5 \\ 0,5 & 1 & 0,4 \\ 0,4 & 0,6 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= \tanh(e_1) \\ f_2(e_2) &= \tanh(e_2) \\ f_3(e_3) &= \tanh(e_3) \end{aligned} \quad (5.5)$$

Para este caso, optamos por um polinômio monotônico com 5 coeficientes, ou seja, grau nove, o que deu origem a um espaço de busca com 24 parâmetros.

Os valores de SIR para as três fontes foram de 30,7 dB, 28,8 dB e 32,8 dB, respectivamente. Fazendo uma analogia com o cenário anterior, percebemos que a SIR, no cenário de três fontes, foi um pouco inferior. Atribuímos essa diminuição principalmente à maior complexidade do problema de busca subjacente, que decorre principalmente do já referido aumento na dimensão do espaço de busca.

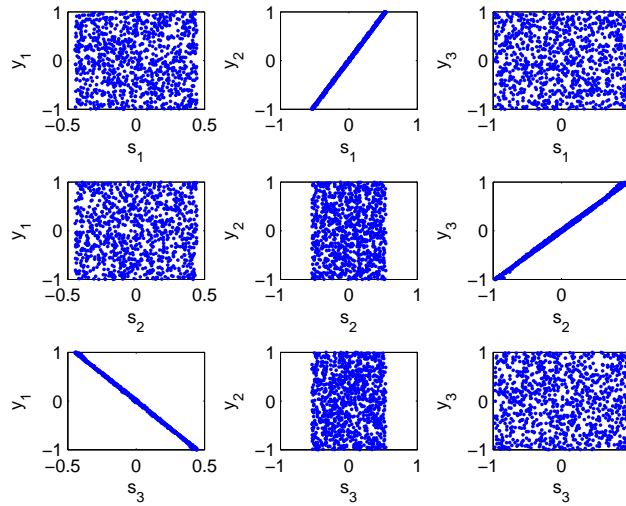


Figura 5.5: Estimativas de cada uma das três fontes.

Visualmente, podemos sumarizar os resultados obtidos conforme expostos na Fig. 5.5, que traz as diversas distribuições conjuntas de fontes e estimativas. Ao observarmos a primeira linha dessa figura, notamos que a fonte estimada y_2 e a fonte s_1 são bem correlacionadas, o que indica que a estimativa aproxima a fonte s_1 . Verificando a segunda e terceira linhas, respectivamente, temos que y_3 aproxima a fonte s_2 e que y_1 é uma boa estimativa de s_3 . Ainda de acordo com a Fig. 5.5, observamos que a última estimativa está invertida em relação à fonte s_3 , o que está de acordo com as ambiguidades previstas na Seção 2.7.1.

5.3 Resultados – Rede Monotônica de Sill

Ao contrário do que ocorreu para os polinômios, os primeiros resultados para a rede neural de Sill não foram satisfatórios. Isso nos levou a realizar uma série de testes adicionais, os quais serão descritos em detalhe a seguir.

Primeiros Testes (Caso Não-Supervisionado)

Para avaliar o desempenho da Rede Monotônica de Sill, optamos inicialmente pela separação de duas fontes uniformemente distribuídas no intervalo $[-1,1]$, e, como nos casos anteriores, misturadas pelo seguinte sistema PNL:

$$A = \begin{bmatrix} 1 & 0,6 \\ 0,5 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= \tanh(e_1) \\ f_2(e_2) &= \tanh(e_2) \end{aligned} \quad (5.6)$$

Neste caso, os parâmetros do algoritmo CLONALG também foram ajustados com base em experimentos preliminares, o que deu origem à Tab. 5.9.

Tabela 5.9: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	250
Número de Clones (N_c)	30
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

No caso da Rede Monotônica de Sill, é preciso determinar a quantidade de grupos e de hiperplanos para que a rede seja capaz de inverter as não-linearidades do sistema misturador. Para este caso, após vários testes, chegou-se a um número de 7 grupos e de 7 hiperplanos, que leva a um espaço de busca de dimensão bem mais elevada que a verificada para o caso dos polinômios: há 200 parâmetros. Isso justifica a maior demanda computacional do algoritmo de otimização, expressa na Tab. 5.9.

Na Fig. 5.6, apresentamos a distribuição conjunta dos dados observados (misturados) e após a separação, para 1000 amostras.

Analisando esta figura, percebe-se que a não-linearidade residual é significativa principalmente nas bordas da distribuição, o que revela que a estrutura do sistema separador, para este caso, não foi capaz de inverter totalmente a ação do sistema misturador.

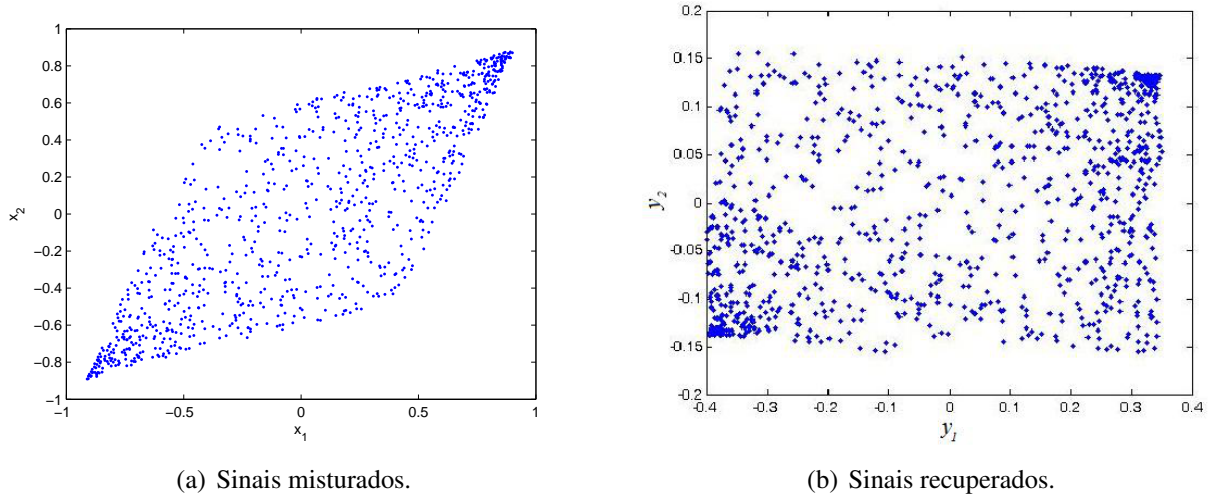


Figura 5.6: Distribuição dos sinais (a) observados e (b) recuperados.

Outro fator que reforça essa conclusão, pode ser visto na Fig. 5.7, que traz a composição das funções não-lineares f_i e g_i . Nota-se, principalmente na Fig. 5.7(b), que a estrutura da rede monotônica de Sill não inverteu completamente as tangentes hiperbólicas. Essas dificuldades são ainda confirmadas pelos valores de SIR para esta rede, que foram baixos, de 11,7 dB e 10,5 dB.

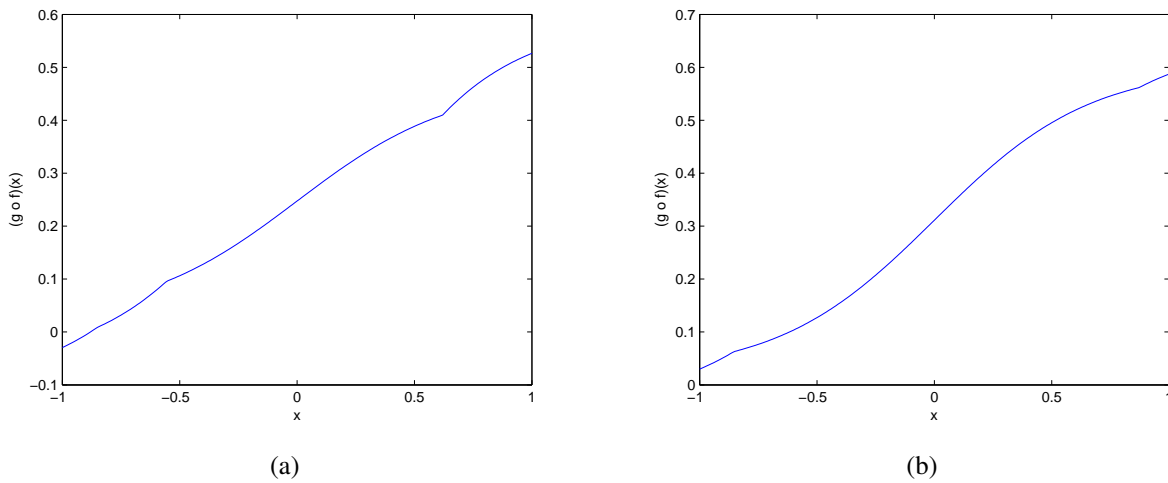


Figura 5.7: Composição das funções f_i e g_i para as duas fontes.

Tendo visto os resultados pouco satisfatórios desta rede no modo não-supervisionado, optamos por investigar seu desempenho num modo de treinamento supervisionado, conforme descrito a seguir.

Caso Supervisionado

Para investigarmos a capacidade estrutural da rede monotônica de Sill no âmbito do modelo PNL, fizemos alguns experimentos num contexto supervisionado, em que utilizamos o erro quadrático médio como função custo. A utilização desta métrica tem o fim de verificar se a rede é estruturalmente capaz de inverter as não-linearidades $f(\cdot)$, retirando da análise, até certo ponto, dificuldades advindas da natureza do critério não-supervisionado. O cenário de teste para este caso foi composto pelo mesmo sistema misturador empregado no caso anterior, mas os parâmetros do algoritmo CLONALG sofreram algumas modificações, conforme mostra a Tab. 5.10.

Tabela 5.10: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	300
Número de Clones (N_c)	50
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

Neste caso, tanto o número de grupos quanto o de hiperplanos foram iguais a 8, valor atingido mais uma vez por meio de experimentos preliminares. A quantidade de parâmetros a serem ajustados é de 260, e, em consequência disso, houve um aumento da quantidade de indivíduos no algoritmo de forma a termos mais condições de explorar o espaço de busca. Isso, mais uma vez, significa uma maior complexidade computacional.

Para essas configurações, a Fig. 5.8 apresenta a distribuição conjunta dos dados observados (misturados) e dos dados obtidos após a separação, tendo sido geradas 1000 amostras dos sinais observados.

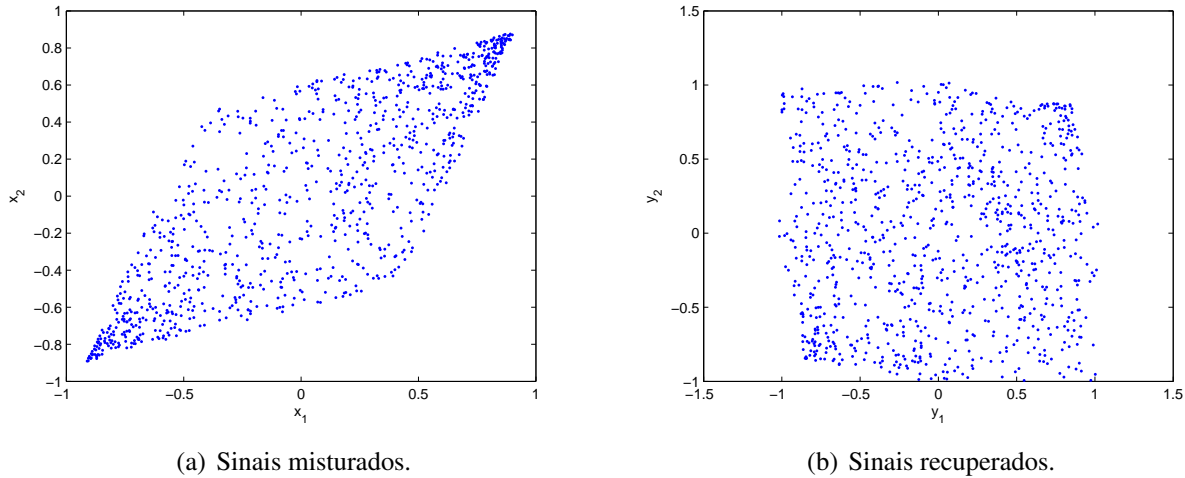


Figura 5.8: Distribuição dos sinais (a) observados e (b) recuperados.

Na Fig. 5.9, apresentamos a composição das funções não-lineares f_i e g_i para esta simulação, sendo perceptível que, apesar de uma melhora relativamente à Fig. 5.7, o mapeamento resultante ainda não está suficientemente próximo de um mapeamento linear, o que revela que a rede monotônica do Sill não foi capaz de inverter totalmente as não-linearidades aplicadas no sistema misturador mesmo no caso supervisionado. Outro fator que aponta para uma melhora de resultado para este caso é o par de valores de SIR, que, para este caso, foi de 25,7 dB para uma fonte e 26,2 dB para a outra. Esta melhora nos resultados parece indicar que os problemas não são estruturais, mas sim relativos ao processo de otimização. Decidimos então diminuir a dimensão do problema fixando a diagonal da matriz de separação \mathbf{W} , o que favorece a abordagem de otimização, ou seja, ao restringirmos a diagonal da matriz de separação, podemos, no melhor caso, inverter a matriz de mistura \mathbf{A} a menos de um ganho de escala. No entanto, como já visto na Seção 2.7.1 há justamente uma ambiguidade de escala em problemas BSS e, assim sendo, não há qualquer perda de generalidade na simplificação considerada. Os resultados, para esse caso, serão apresentados a seguir.

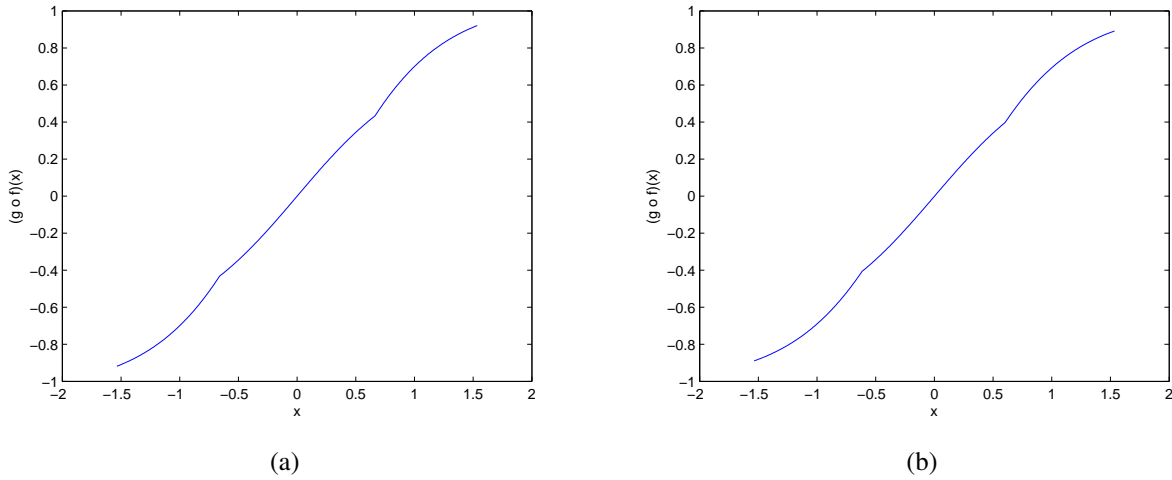


Figura 5.9: Composição das funções f_i e g_i para as duas fontes.

Resultado Final (Caso Não-Supervisionado)

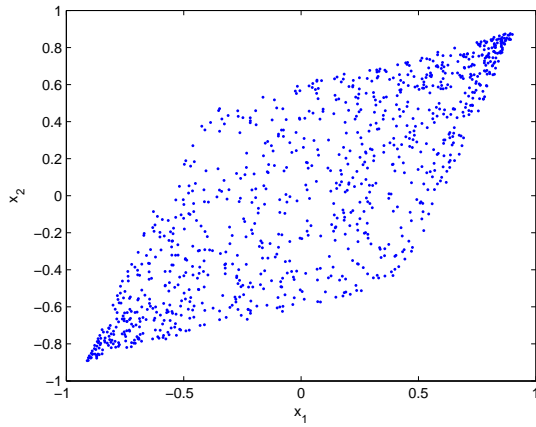
Para avaliar o desempenho da rede monotônica de Sill fixando parte da matriz de separação, mantivemos os modelos de fonte e mistura usados nos casos anteriores. Os parâmetros do algoritmo CLONALG foram os mesmos do caso supervisionado, conforme a Tab. 5.11. Mas, neste caso, o número de grupos foi de 6 e o número de hiperplanos foi também de 6, reduzindo dessa forma a dimensão do espaço de busca para 148 parâmetros. Isto simplifica o problema de busca e reduz a demanda computacional por parte do algoritmo de otimização. Estes valores de grupos e hiperplanos foram ajustados iniciando com o menor valor possível, que é 2, e realizando um aumento gradativo até encontrar um resultado satisfatório em termos da recuperação das fontes.

Tabela 5.11: Parâmetros do Algoritmo CLONALG

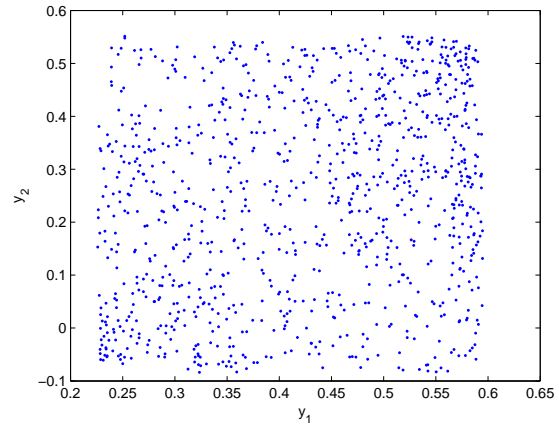
Parâmetro	Valor
Número de Indivíduos (N)	300
Número de Clones (N_c)	50
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

Na Fig. 5.10, mostra-se a distribuição conjunta dos dados observados (misturados) e dos dados após a separação para 1000 amostras dos sinais observados. Já na Fig. 5.11, apresentamos a composição das funções não-lineares f_i e g_i obtidas nesta simulação. Os valores de SIR para as duas

fontes, neste caso, foram de 15,7 dB e 23,5 dB.

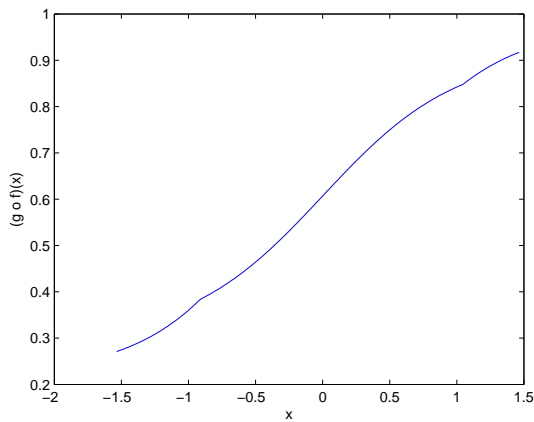


(a) Sinais misturados.

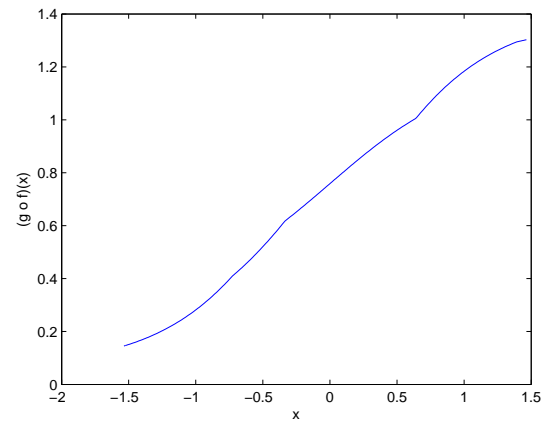


(b) Sinais recuperados.

Figura 5.10: Distribuição dos sinais (a) observados e (b) recuperados.



(a)



(b)

Figura 5.11: Composição das funções f_i e g_i para as duas fontes.

Os resultados, neste caso, foram melhores que os obtidos no caso não-supervisionado original, o que indica, mais uma vez, que as dificuldades verificadas no uso da rede neural de Sill advêm da etapa de otimização.

5.4 Resultados – Rede Neural MLP Monotônica

Para avaliar o desempenho da rede neural MLP monotônica, foi preciso adotar experimentos semelhantes aos relatados na seção 5.3, pois as dificuldades encontradas no modo não-supervisionado foram novamente verificadas. O conjunto de testes realizados será exposto a seguir.

Primeiros Testes (Caso Não-Supervisionado)

Adotaremos, mais uma vez, fontes uniformemente distribuídas no intervalo $[-1,1]$ e o seguinte modelo de mistura ³:

$$A = \begin{bmatrix} 1 & 0,6 \\ 0,5 & 1 \end{bmatrix} \quad \text{e} \quad \begin{aligned} f_1(e_1) &= \tanh(e_1) \\ f_2(e_2) &= \tanh(e_2) \end{aligned} \quad (5.7)$$

Os parâmetros escolhidos para o algoritmo CLONALG são mostrados na Tab. 5.12.

Tabela 5.12: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	200
Número de Clones (N_c)	40
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

No caso da rede MLP Monotônica temos que estipular a quantidade de tangentes hiperbólicas da primeira camada e da segunda camada para que a rede seja capaz de inverter as não-linearidades do sistema misturador. Após uma série de testes, o número de tangentes hiperbólicas da primeira camada foi ajustado para 4 e o número de tangentes hiperbólicas da segunda camada foi ajustado para o mesmo valor, o que deu origem a um espaço de busca com 116 parâmetros.

A Fig. 5.12 apresenta a distribuição conjunta dos dados observados (misturados) e dos dados após a separação, para 1000 amostras dos sinais observados, enquanto, na Fig. 5.13, apresentamos a composição das funções não-lineares f_i e g_i obtidas nesta simulação.

³Note que se trata, a princípio, de um problema de difícil tratamento por parte da rede neural, uma vez que a não-linearidade a ser invertida é da mesma natureza das funções por ela usadas para compor mapeamentos

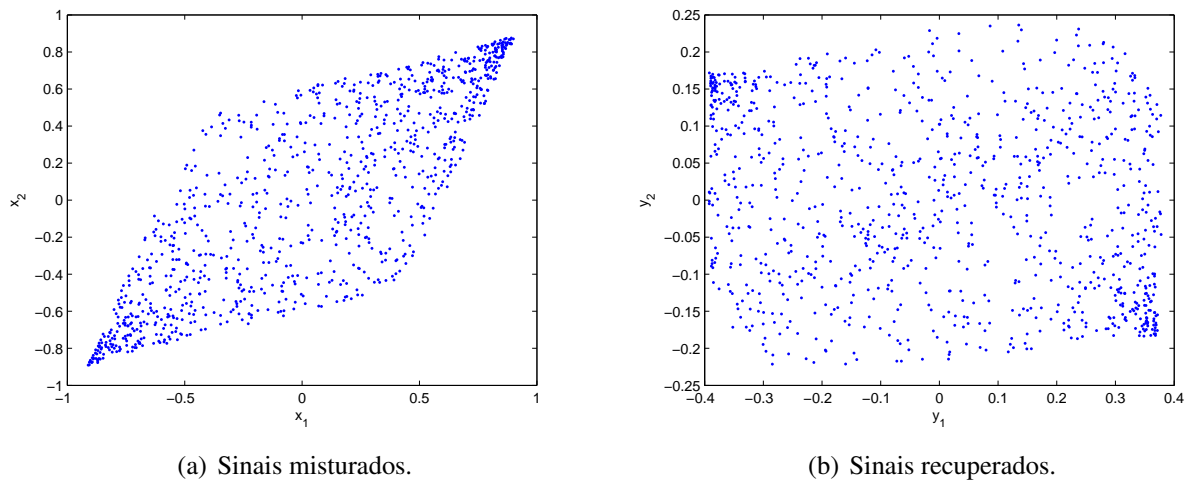


Figura 5.12: Distribuição dos sinais (a) observados e (b) recuperados.

Percebe-se na Fig 5.12(b) que o algoritmo não foi capaz de eliminar o efeito das não-linearidades do sistema misturador, o que também é confirmado pela Fig. 5.13 e pelos valores de SIR, que foram de 13,2 dB e 10,5 dB. Acreditamos que, novamente, esse desempenho pouco satisfatório se deveu a dificuldades relativas ao processo de otimização, ou seja, a um fator não estrutural. Para averiguar esse fato, realizamos um teste num caso supervisionado.

Caso Supervisionado

O desempenho da rede neural MLP monotônica, neste caso, foi avaliado segundo a mesma metodologia exposta na seção 5.3. Os parâmetros do algoritmo CLONALG está na Tab. 5.13:

Tabela 5.13: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	300
Número de Clones (N_c)	50
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

Neste caso, mais uma vez o número de tangentes hiperbólicas na primeira camada foi de 4, assim como o número de tangentes hiperbólicas na segunda camada. Estes valores foram ajustados de maneira análoga à exposta para o caso da rede monotônica de Sill, iniciando do menor valor

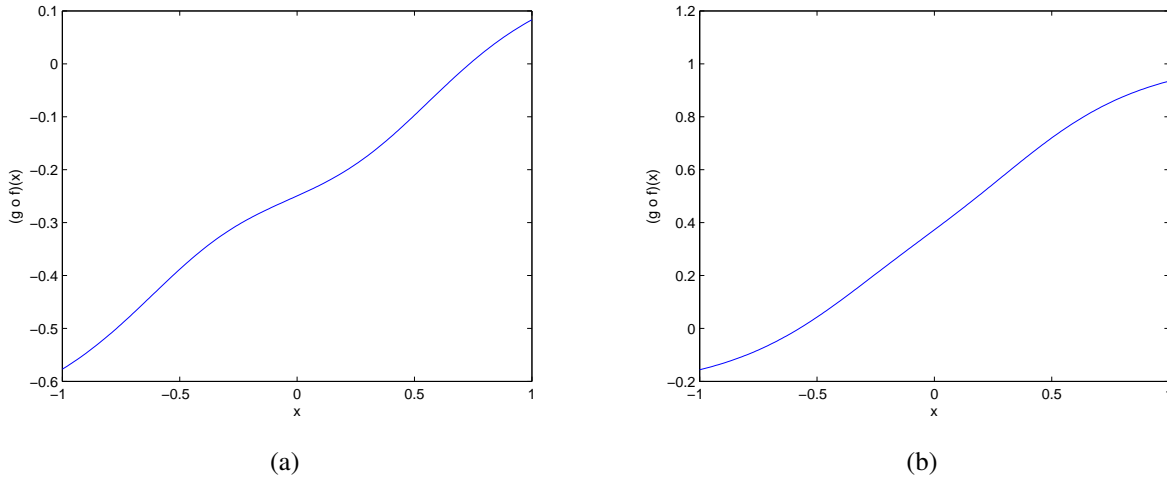


Figura 5.13: Composição das funções f_i e g_i para as duas fontes.

possível até encontrar resultados satisfatórios. Como não houve modificações na quantidade de tangentes hiperbólicas da primeira e segunda camadas em comparação com o caso anterior, o número de parâmetros a serem ajustados permanece em 116. O aumento na quantidade de indivíduos para este caso se justifica, portanto, exclusivamente do ponto de vista de melhorar o processo de busca.

A Fig. 5.14 apresenta a distribuição conjunta dos dados observados (misturados) e dos dados após a separação, para 1000 amostras dos sinais observados, enquanto na Fig. 5.15, apresentamos a composição das funções não-lineares f_i e g_i obtidas nesta simulação.

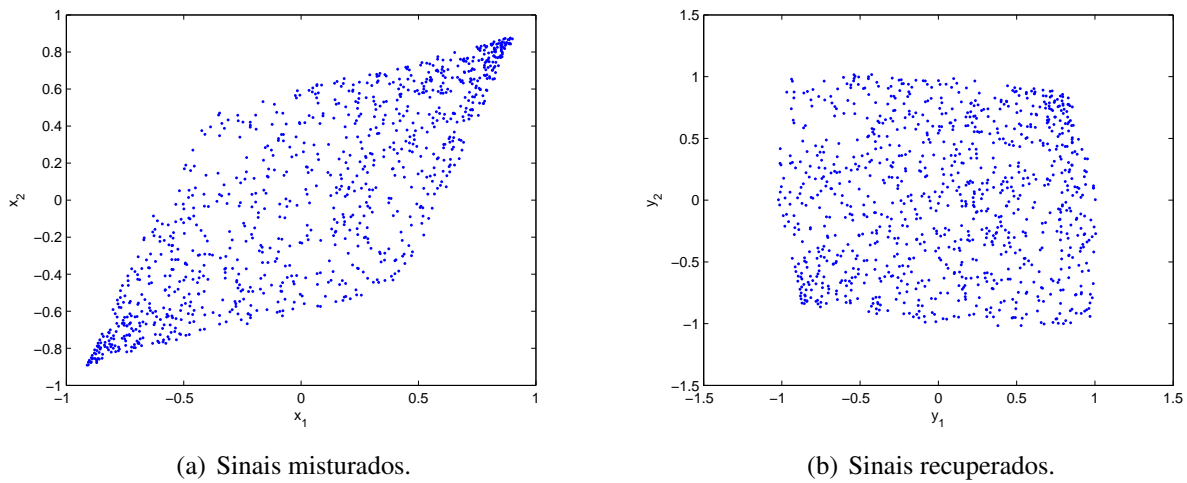


Figura 5.14: Distribuição dos sinais (a) observados e (b) recuperados.

Neste caso, houve uma substancial melhora em comparação com o experimento no caso não-supervisionado, que é confirmado pelos valores de SIR das fontes, que foram de 27,3 dB e 28,5 dB

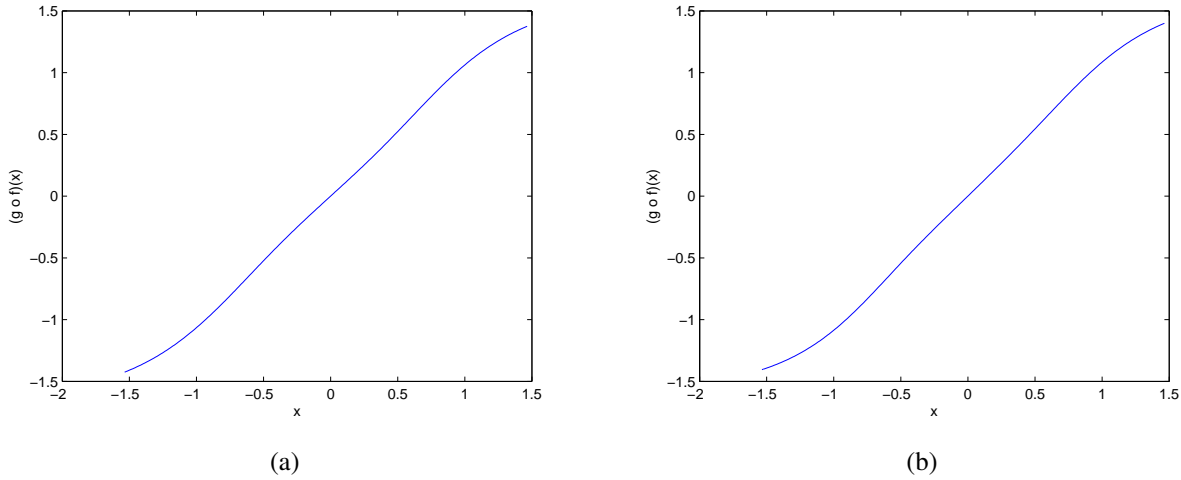


Figura 5.15: Composição das funções f_i e g_i para as duas fontes.

respectivamente.

Os resultados confirmam que, do ponto de vista estrutural, a rede é capaz de inverter satisfatoriamente as não-linearidades do modelo de mistura, o que indica que as dificuldades anteriores decorreram da otimização da informação mútua.

Resultados Finais (Caso Não-Supervisionado)

Para testar a validade dessa conclusão, retornaremos a um modo de operação não-supervisionado, no mesmo cenário, mas com a diagonal principal da matriz de separação fixa. Os parâmetros do algoritmo CLONALG foram ajustados conforme a Tab. 5.14:

Tabela 5.14: Parâmetros do Algoritmo CLONALG

Parâmetro	Valor
Número de Indivíduos (N)	300
Número de Clones (N_c)	50
Regulador de Amplitude de Mutação (β)	60/120
Número de Indivíduos Reinseridos	20
Número Máximo de Gerações	10000

O número de tangentes hiperbólicas na primeira e na segunda camada foi de 3. Como no caso da rede monotônica de Sill, há uma redução na dimensão do espaço de busca, desta vez, para 70 parâmetros. Deste modo, a complexidade computacional diminuiu e a simulação tornou-se mais

rápida em comparação com os outros dois casos anteriores.

Temos, na Fig. 5.16, a distribuição conjunta dos dados observados (misturados) e dos dados após a separação, para 1000 amostras, e, na Fig. 5.17 apresentamos a composição das funções não-lineares f_i e g_i obtidas.

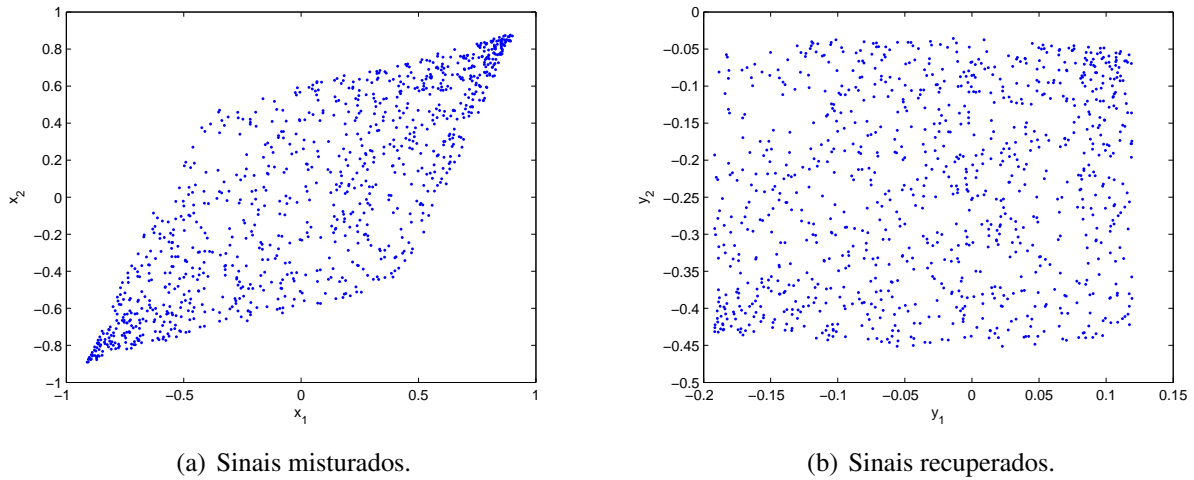


Figura 5.16: Distribuição dos sinais (a) observados e (b) recuperados.

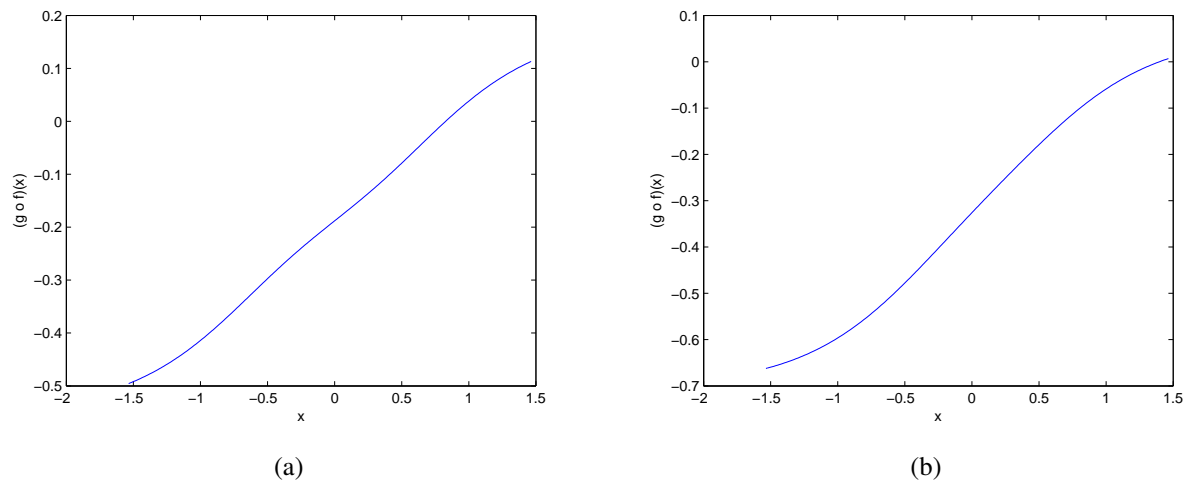


Figura 5.17: Composição das funções f_i e g_i para as duas fontes.

Os resultados mostraram uma significativa melhora em comparação com o primeiro teste não-supervisionado desta rede, que também é confirmado pelos valores de SIR obtidos para as duas fontes, que foram de 24,1 dB e 15,4 dB. Isso dá suporte adicional à hipótese de que a otimização da informação mútua, no primeiro caso, não foi plenamente satisfatória devido, provavelmente, à dimensão do espaço de busca. Isso indica claramente, como exporemos no capítulo final, que há espaço para a investigação do uso de técnicas mais elaboradas de otimização no contexto deste problema.

Capítulo 6

Conclusões e Perspectivas

Neste trabalho, foram apresentadas duas contribuições ao estudo do problema de separação cega de fontes no contexto de misturas com não-linearidade posterior. A primeira delas foi uma análise comparativa de duas ferramentas bio-inspiradas de otimização - uma versão do algoritmo CLONALG para lidar com valores reais e um algoritmo de otimização de enxame de partículas - no âmbito do modelo ad hoc de separação apresentado em Duarte et al. [1]. Essa análise mostrou, em termos simples, que ambas as técnicas empregadas são eficientes na otimização da métrica de informação mútua, tendo o CLONALG, no entanto, apresentado maior robustez. A segunda contribuição foi a proposta de uma metodologia geral para separação de misturas PNL, metodologia esta baseada na própria combinação CLONALG + função custo de informação mútua e em estruturas monotônicas. A ideia foi testada para três modelos não-lineares - um aproximador polinomial, a rede neural monotônica de Sill e uma rede MLP monotônica - em cenários de diferente complexidade. Os testes indicaram a solidez da proposta, especialmente quando se empregam polinômios com restrição de monotonicidade, que dão origem a um interessante compromisso entre complexidade (em termos de número de parâmetros livres) e desempenho. As demais estruturas também foram capazes de levar a condições de separação, embora com menos eficiência, o que se explica devido à dimensão do espaço de busca por elas definido.

Como perspectivas para trabalhos futuros, apontamos principalmente a análise de desempenho da proposta com o emprego de ferramentas de otimização mais elaboradas e eficientes, que podem levar a um emprego efetivo da capacidade de aproximação das redes neurais. Também pretendemos testar o paradigma proposto em cenários práticos, principalmente no que diz respeito a aplicações em arranjo de sensores químicos [6], fazer uma comparação de desempenho sistemática com a proposta original de Taleb e Jutten [8] e analisar detalhadamente as características da função custo de informação mútua.

Referências Bibliográficas

- [1] L. T. Duarte, R. Suyama, R. R. F. Attux, F. J. Von Zuben, and J. M. T. Romano. Blind Source Separation of Post-nonlinear Mixtures Using Evolutionary Computation and Order Statistics. In *Independent Component Analysis and Blind Signal Separation*, pages 66–73, 2006.
- [2] L. T. Duarte. *Um Estudo sobre Separação Cega de Fontes e Contribuições ao Caso de Misturas Não-lineares*. Tese de Mestrado, Universidade Estadual de Campinas (Brasil), agosto 2006.
- [3] L. N. de Castro and J. I. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, 2002.
- [4] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.
- [5] S. C. Douglas and S. Haykin. On the Relationship between blind deconvolution and blind separation. In S. Haykin, editor, *Unsupervised Adaptive Filtering*, volume 2, pages 1591–1595, California, EUA, 2000. Wiley.
- [6] L. T. Duarte. *Design of Smart Chemical Sensor Arrays: an Approach Based on Source Separation Methods*. Tese de Doutorado, Institut National Polytechnique de Grenoble (França), novembro 2009.
- [7] P. Comon. Independent Component Analysis, a New Concept? *Signal Processing*, 36(6):287–314, 1994.
- [8] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820, 1999.
- [9] A. Larue, C. Jutten, and S. Hosseini. Markovian source separation in post-nonlinear mixtures. In *Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2004*, pages 702–709, 2004.
- [10] Joseph Sill. Monotonic networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 661–667, 1998.

- [11] Bernhard Lang. Monotonic Multi-layer Perceptron Networks as Universal Approximators. In *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005*, pages 31–37, 2005.
- [12] L. N. de Castro and F. J. Von Zuben. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, 6(3):239–251, 2002.
- [13] D.-T. Pham. Blind Separation of Instantaneous Mixtures of Sources Based on Order Statistics. *IEEE Transactions on Signal Processing*, 48(2):363–375, 2000.
- [14] J. Hérault, C. Jutten, and B. Ans. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In *Actes du Xème Colloque GRETSI*, pages 1017–1022, Nice, France, 1985.
- [15] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical*, 27:379–423, 623–656, 1948.
- [16] G. Darmois. Analyse Générale des Liasons Stochastiques. *Revue de l’Institut International de Statistique*, 21:2–8, 1953.
- [17] J. F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.
- [18] J. F. Cardoso and A. Souloumiac. Blind Beamforming for non-Gaussian Signals. *IEEE Proceedings Radar and Signal Processing*, 140(6):362–370, 1993.
- [19] J. F. Cardoso and B. H. Laheld. Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017–3030, 1996.
- [20] A. J. Bell and T. J. Sejnowski. An Information Maximisation Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.
- [21] J. Karhunen, P. Pajunen, and E. Oja. The Nonlinear PCA Criterion in Blind Source Separation: Relations with Other Approaches. *Neurocomputing*, 22:5–20, 1998.
- [22] B. Arons. A Review of the Cocktail Party Effect. *Journal of the American Voice I/O Society*, 12:35–50, 1992.
- [23] A. Tonazzini, I. Gerace, and F. Martinelli. Multichannel Blind Separation and Deconvolution of Images for Document Analysis. *IEEE Transactions on Image Processing*, 19(4):912–925, 2010.

- [24] S. U. H. Qureshi. Adaptive Equalization. In *Proceedings of the IEEE*, pages 1349–1387, 1985.
- [25] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, 3 edition, 1996.
- [26] R. R. F. Attux, A. Neves, L. T. Duarte, R. Suyama, C. C. M. Junqueira, L. E. P. Rangel, T. M. Dias, and J. M. T. Romano. On the Relationships between Blind Equalization and Blind Source Separation - Part II: Relationships. *Journal of the Brazilian Telecommunications Society*, 2006.
- [27] S. Bermejo, C. Jutten, and J. Cabestany. ISFET Source Separation: Foundations and Techniques. *Sensors and Actuators B - Chemical*, 13:222–233, 2006.
- [28] E. Kofidis. Blind source separation: Fundamentals and recent advances. Technical report, Mini-curso no XIX Simpósio Brasileiro de Telecomunicações (SBrT2001), 2001.
- [29] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 3 edition, 1993.
- [30] S. Haykin. *Blind Deconvolution*. Prentice-Hall, 1994.
- [31] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc, 2 edition, 1991.
- [32] J. F. Cardoso. Infomax and Maximum Likelihood of Blind Source Separation. *IEEE Signal Processing Letters*, 4(4):112–114, 1997.
- [33] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, 1988.
- [34] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.
- [35] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks: Theory and Applications*. John Wiley, 1996.
- [36] C. Jutten and J. Karhunen. Advances in Nonlinear Blind Source Separation. In *Proceedings of the Fourth International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2003*, Nara, Japão, 2003.
- [37] A. Hyvärinen and P. Pajunen. Nonlinear Independent Component Analysis: Existence and Uniqueness Results. *Neural Networks*, 3:429–439, 1999.
- [38] J. Eriksson and V. Koivunen. Blind identifiability of class of nonlinear instantaneous ICA models. In *Proceedings of the XI European Signal Processing Conference (EUSIPCO 2002)*, volume 2, pages 7–10, Toulouse, France, setembro 2002.

- [39] M. Babaie-Zadeh. *On Blind Source Separation in Convolutional and Nonlinear Mixtures*. Tese de Doutorado, Institut National Polytechnique de Grenoble (França), setembro 2002.
- [40] F. Rojas, C. G. Puntonet, M. Rodríguez-Álvarez, I. Rojas, and R. Martín-Clemente. Blind Source Separation in Post-Nonlinear Mixtures Using Competitive Learning, Simulated Annealing, and a Genetic Algorithm. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 34(4):407–416, 2004.
- [41] L. N. de Castro. *Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications*. CRC Press, 1 edition, 2006.
- [42] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001.
- [43] F. M. Burnet. *The Clonal Selection Theory of Acquired Immunity*. Cambridge University Press, 1959.
- [44] J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, Australia, 1995.
- [45] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Hardcover, 2 edition, 1998.